

І. НЕВЛЮДОВ, С. НОВОСЕЛОВ, К. СУХАЧОВ

## МЕТОД ОДНОЧАСНОЇ ЛОКАЛІЗАЦІЇ ТА КАРТОГРАФУВАННЯ ДЛЯ ПОБУДОВИ 2,5D-КАРТИ НАВКОЛИШНЬОГО СЕРЕДОВИЩА ЗАСОБАМИ ROS

Метод *SLAM* (одночасної локалізації та картографування) на сьогодні є актуальною темою для досліджень і розвитку в галузі робототехніки та комп'ютерного зору. *SLAM* широко застосовується в різних сферах, зокрема автономної навігації інтелектуальних роботів. З допомогою цього методу розв'язуються проблеми в розширеній і віртуальній реальності, БПЛА та інших систем. За останні роки *SLAM* здобув значні досягнення завдяки поступовому розвитку його алгоритмів, використанню новітніх датчиків, а також покращенню обчислювальної потужності комп'ютерів. **Предметом дослідження** є сучасні методи одночасної локалізації та картографування в режимі реального часу. **Мета роботи** – моделювання розробленого алгоритму для побудови карт навколишнього середовища та визначення місця розташування й орієнтації інтелектуального робота в просторі в режимі реального часу за допомогою пакетів ROS. **Завдання статті** – демонстрація результатів поєднання методів *SLAM* та розроблення нових підходів до розв'язання проблем одночасної локалізації та картографування. Для досягнення поставлених завдань використано комбінацію методів лазерного сканування (*2D LRF*) та глибинного відтворення зображень (*RGB-D*) для одночасної локалізації та картографування інтелектуального робота та побудови 2,5D-карти середовища. Здобуті результати є обнадійливими та демонструють перспективність роботи об'єднаних методів *SLAM*, що застосовуються разом для забезпечення й очного виконання одночасної локалізації та картографування інтелектуальних роботів у режимі реального часу. Запропонований метод дає змогу враховувати висоти перешкод у побудові карти навколишнього середовища, витрачаючи менші обчислювальні потужності. У висновку такий підхід розширює технології, не замінюючи наявні робочі пропозиції, й уможливує використання сучасних методів для всебічного виявлення та розпізнавання довкілля за допомогою ефективного локалізаційного та картографічного підходу, надаючи більш точні результати з використанням менших ресурсів.

**Ключові слова:** *SLAM*; ROS; *2D LRF*; *RGB-D*; 2,5D-карта висоти; методи одночасної локалізації та картографування; інтелектуальний робот; визначення місця розташування; моделювання та симуляція.

### Вступ

Роботи є передовою технологією, що має значний вплив на різні сфери суспільства. Вони є універсальними пристроями для відтворення рухових та інтелектуальних функцій і можуть виконувати види діяльності, які для людини трудомісткі, тяжкі, монотонні, шкідливі для її здоров'я та життя.

Незважаючи на те, що більшістю роботів досі керують фахівці й ці технології працюють за чіткою програмою, з'являється все більше інтелектуальних пристроїв, що можуть виконувати різноманітні складні операції без утручання людини [1].

Особливо актуальним є розроблення інтелектуальних роботів із розвиненими можливостями одночасно локалізувати та картографувати навколишнє середовище в режимі реального часу. Це дає змогу здійснювати автономне функціонування інтелектуальних мобільних роботів, зокрема пересування в незнайомій місцевості, у незвичних умовах і за наявності певної міри невизначеності даних, що надходять від довкілля.

З огляду на сказане для забезпечення автономного функціонування роботів, зокрема їх пересування в незнайомих місцевостях та в складних умовах, надзвичайно важливе значення набувають системи розміщення, зокрема їх методи локалізації та картографування.

Нині більшість автономних роботів використовують своє відображення навколишнього середовища у 2D- або 3D-картах для того, щоб точно визначити своє місце розташування та безпечно переміщуватися у фізичному просторі. Однак однією з основних проблем картографування інтелектуальних роботів полягає в тому, що 2D-карта не містить достатньої інформації про перешкоди на різних рівнях висот, а для створення 3D-карт потрібні значні обчислювальні потужності, багато часу на оброблення даних і значний обсяг пам'яті.

У цій роботі пропонується метод локалізації інтелектуального робота за допомогою співпраці методів лазерного сканування (*2D LRF*) та глибинного відтворення зображень (*RGB-D*) для побудови 2,5D-карти [2].

Такий підхід є особливо корисним у ситуаціях, коли інтелектуальний мобільний робот має сприймати інформацію про перешкоди на різних рівнях висот. Багато сучасних автономних систем стикаються з такими сценаріями, що обмежують їх можливості для розгортання та збільшують витрати, оскільки середовище має бути ретельно підготовлено, щоб вони могли функціонувати.

Ефективність запропонованого методу була перевірена за допомогою симуляції в програмному середовищі *ROS (Robot Operating System)*. Унаслідок цього згенеровано *2,5D*-карту висот, що була використана для автономної навігації та генерування маршруту руху інтелектуального робота.

### Аналіз останніх досліджень і публікацій

Інтелектуальний мобільний робот може визначати своє місце розташування за допомогою апріорно наявної карти простору або шляхом аналізу своїх спостережень [3]. В ідеальному випадку карту навколишнього середовища було б добре завантажити в робота завчасно, але на практиці це не завжди можливо.

Тому постає завдання навчити робота одночасно будувати карту довкілля та визначати своє місце розташування в цьому просторі, розробляючи можливий шлях траєкторії переміщення.

Напрямок, що описує методи розв'язання цього завдання, називається *SLAM (Simultaneous Localization And Mapping)*. *SLAM* – це завдання у сфері робототехніки, що потребує від робота одночасно визначати своє місце в середовищі й будувати мапу цього середовища [4].

Методи одночасної локалізації та картографування (*SLAM*) – це концепція, яка сполучає два взаємно залежних процеси навігації та побудови карт в єдиний цикл обчислень. Це дає змогу інтелектуальним роботам отримувати дані про навколишнє середовище, створювати карту, визначати своє місце розташування та орієнтацію в просторі.

Сутність методу полягає в тому, що робот використовує різні датчики (наприклад, камеру, лазерний далекомір, гіроскоп та акселерометр) для збирання інформації про довкілля та своє розміщення. Потім він обробляє ці дані, щоб зрозуміти, як він перебуває в просторі та як навколишнє середовище виглядає.

Варто зазначити, що *SLAM* – це не якийсь конкретний алгоритм, а набір методів і різноманітних засобів, що допомагають виконувати завдання щодо визначення місця розташування робота та побудови карти місцевості. На сьогодні існує значна кількість методів, що використовують як апаратні, так і програмні можливості. Вибір методу залежить від конкретного завдання та характеристик застосовуваних датчиків. Однак загальною метою цих методів є забезпечення високої точності та швидкості оброблення інформації з датчиків, щоб дати змогу роботу зорієнтуватися в навколишньому просторі та рухатися в ньому безпечно та ефективно.

Основна ідея більшості методів *SLAM* та їх алгоритмів проста. Перебуваючи в певному положенні, робот починає виявляти об'єкти навколо. Зробивши перші заміри відстаней, він їх запам'ятовує та рухається в напрямку інших об'єктів. Після того, як усі об'єкти в деякому просторі знайдені та відстані до них виміряні, робот повертається на вихідну позицію. Швидше за все, він потрапить в інше місце, а не в те, у якому перебував спочатку. Відбувається це через помилки одометрії. Далі робот починає друге коло вимірювань, у якому помилка скорочується.

Нині найбільш популярними є методи, основані на розширеному фільтрі Калмана (*EKF*) [5] та на фільтрі частинок (метод Монте-Карло) [6]. Поступово другий підхід витісняє перший. Основним недоліком розширеного фільтра Калмана є його квадратична обчислювальна складність від кількості об'єктів на карті. Фільтр частинок має логарифмічну складність (залежить від кількості осередків на карті та кількості частинок).

Під час розв'язання завдання *SLAM* постають дві основні проблеми. Перша – це проблема збіжності. Вона безпосередньо пов'язана з точністю обчислень. Будь-які датчики та системи одометрії мають певну модель помилки. Проте точно визначити цю модель найчастіше неможливо, тому користуються різноманітними спрощеннями, що спричиняють неточності в побудові карти.

Друга проблема – обчислювальна складність алгоритмів. Частково цю проблему вирішено, й на сьогодні існують алгоритми, що асимптотично розв'язують це завдання за логарифмічний час. Однак структурна складність довкілля така, що навіть за таких обчислень не завжди вдається вирішувати завдання в прийнятний час.

Додатковою проблемою є те, що завдання *SLAM* найчастіше постає перед інтелектуальними роботами, для яких питання споживання електроенергії є найважливішим, отже, з метою енергозбереження розробники інтелектуальних роботів змушені обмежувати обчислювальні потужності використовуваних апаратних платформ.

Зрештою, незважаючи на те, що очі робота (його сенсори) бачать об'єкти в темряві, він не завжди може впоратися з обсягом і якістю інформації, яка надходить.

### Математична модель проблеми *SLAM*

*SLAM* – це проблема оцінювання. Ми хочемо оцінити як змінну  $X$ , що містить траєкторію або позу робота, так і змінну  $M$ , що подає положення орієнтирів у середовищі. З огляду на набір вимірювань  $Z = \{z_1, \dots, z_m\}$  і модель вимірювання або спостереження  $h(\cdot)$ , що виражає  $z_k$  як функцію  $X$  і  $M$  [5], маємо

$$z_k = h(X_k, M_k) + \epsilon_k, \quad (1)$$

де  $X_k, M_k$  – підмножини  $X$  та  $M$ , а  $\epsilon_k$  – шум випадкового вимірювання.

*SLAM* має тенденцію розв'язувати проблему максимальної апостеріорної ймовірності (*MAP*) таким чином, що:

$$\begin{aligned} \{X^*, M^*\} &= \operatorname{argmax}_{\{X, M\}} p(X, M | Z) = \\ &= \operatorname{argmax}_{\{X, M\}} p(Z | X, M) p(X, M), \end{aligned} \quad (2)$$

де  $(Z | X, M)$  – це ймовірність вимірювання  $Z$  за  $X$  і  $M$ , а  $p(X, M)$  – це апіорне знання  $X$  та  $M$ .

Якщо припустити, що спостереження  $z_k$  є незалежними, проблема *MAP* виглядає так:

$$\begin{aligned} \{X^*, M^*\} &= \operatorname{argmax}_{\{X, M\}} \prod_{k=1}^m p(z_k | X, M) p(X, M) = \\ &= \operatorname{argmax}_{\{X, M\}} \prod_{k=1}^m p(z_k | X_k, M_k) p(X, M). \end{aligned} \quad (3)$$

### Імовірнісне рішення *SLAM*-фреймворку

Як було зазначено вище, *SLAM* є рекурсивним процесом оцінювання. Такий процес часто розглядається в імовірнісному вигляді, коли потрібно виконувати класичний крок передбачення та оновлення.

Розглядаючи робота, який рухається в невідомому середовищі, визначаємо [5]:

$x_k$  – вектор стану, що описує робота в момент часу  $k$ ;

$x_{k|k-1}$  – оцінений вектор стану в момент часу  $k$ , беручи до уваги знання про попередній стан;

$u_k$  – вектор керування, застосований на  $k-1$  для переміщення транспортного засобу в стан  $x_k$  (якщо він був передбачений);

$m_i$  – вектор, що описує  $i$ -й орієнтир;

$z_{k,i}$  – спостереження  $i$ -ї ознаки, зробленої в час  $k$ ;

$X$  – набір місць розташування транспортних засобів від часу 0 до  $k$ ;

$U_{0:k}$  – набір керувальних входів від часу 0 до  $k$ ;

$Z_{0:k}$  – набір спостережень від часу 0 до  $k$ ;

$M$  – набір орієнтирів або карт;

$M_{k|k-1}$  – оцінювання карти в момент часу  $k$  з огляду на попередню карту в момент часу  $k-1$ .

Оскільки ми розглядаємо ймовірнісну форму *SLAM*, у кожен момент часу  $k$  хочемо обчислити функцію розподілу ймовірностей:

$$P(x_k, M | Z_{0:k}, U_{0:k}). \quad (4)$$

Щоб продовжити, потрібно застосувати рекурсивний метод, який оперує апіорними даними  $P(x_{k-1|k-1}, M_{k-1|k-1} | Z_{0:k-1}, U_{0:k-1})$ , оновлюваними за допомогою  $u_k$  і  $z_k$ .

Для цього спочатку необхідно визначити модель руху, що передбачає стан за вхідним сигналом керування  $P(x_k | x_{k-1}, u_k)$ , таким чином:

$$\begin{aligned} &P(x_{k|k-1}, M_{k|k-1} | Z_{0:k-1}, U_{0:k}) = \\ &= \int P(x_{k|k-1} | x_{k-1|k-1}, u_k) \times \\ &\times P(x_{k-1|k-1}, M_{k-1|k-1} | Z_{0:k-1}, U_{0:k-1}) dx_{k-1|k-1}. \end{aligned} \quad (5)$$

Аналогічно також потрібно визначити модель сприймання або спостереження  $P(z_{i,k} | X_k, M)$ , яка пов'язує інформацію датчика щодо виявлення  $i$  в час  $k$  з оцінкою стану, таким чином:

$$P(x_{k|k}, M_{k|k} | Z_{0:k}, U_{0:k}) = \frac{P(z_{i,k} | x_{k|k-1}, M_{k|k-1})}{P(x_{k|k-1}, M_{k|k-1} | Z_{0:k-1}, U_{0:k})}. \quad (6)$$

Початково проблему максимізації апостеріорної ймовірності в алгоритмі *SLAM* вирішували за допомогою розширеного фільтра Калмана (*EKF*). Він зменшує невизначеність та надає оцінку на кожному кроці алгоритму. Використовуючи

ймовірнісну модель, *EKF* гарантує збіжність та послідовність карти. Однак він дуже чутливий до помилок асоціації даних, а постійне оновлення всіх орієнтирів та їх матриці коваріації потребує значного обчислювального зусилля. На рис. 1 зображено блок-схему процесу *EKF-SLAM*. Сучасні підходи до вирішення проблеми *MAP* використовують оптимізаційні методи, зокрема *Bundle-Adjustment (BA)* [8], або глибокі нейронні мережі [9].



Рис. 1. Блок-діаграма процесу *EKF-SLAM*

Коли екстероцептивні дані надходять у час  $t$ , стан робота на цей момент визначається за допомогою рівняння (5), а виявлені ознаки порівнюються з тими, що є на мапі. Збіги дозволяють оновлювати стан і карту з допомогою рівняння (6). Якщо виявлення немає на мапі, воно ініціалізується, якщо це можливо, і додається до неї. Це виконується рекурсивно.

На рис. 2 показано процес *SLAM* і зображені всі змінні, що використовуються в цьому розділі.

Траєкторія із сірими трикутниками побудована на основі оцінок, а траєкторія із білими трикутниками є "правдивою", побудованою на основі реальних фактів.

На цьому кроці ми обчислили як крок передбачення, так і крок оновлення, і *SLAM* може бути поданий як ітеративна оцінка для наступного рівняння, що є комбінацією (5) і (6):

$$P(x_{k|k}, M_{k|k} | Z_{0:k}, U_{0:k}) = \frac{P(z_{k,i} | x_{k|k-1}, M_{k|k-1}) \times \int P(x_{k|k-1} | x_{k-1|k-1}, u_k)}{P(x_{k-1|k-1}, M_{k-1|k-1} | Z_{0:k-1}, U_{0:k-1})} dx_{k-1|k-1} \quad (7)$$

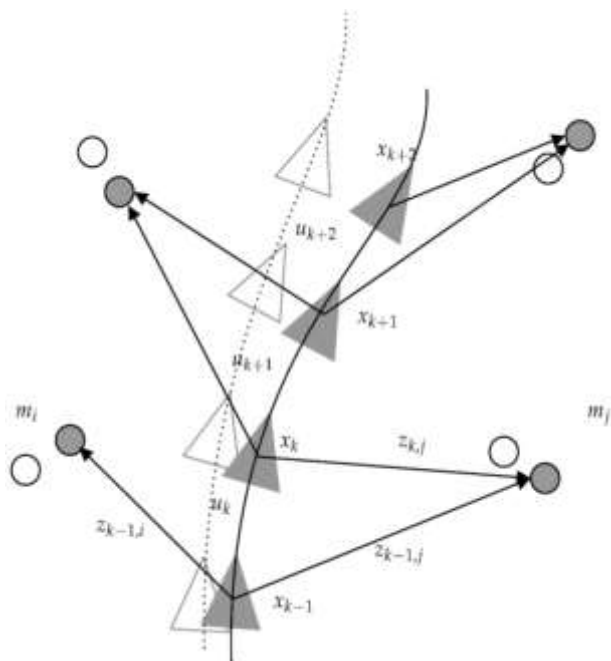


Рис. 2. Ілюстрація процесу *SLAM* та позначки [4]

Рівняння (7) надає рекурсивний байєсівський метод для реалізації *SLAM*. Розв'язання проблеми *SLAM* має обробляти відповідні обчислення для моделі руху й моделі сприйняття, щоб ефективно розрахувати рекурсивний метод. Сучасні підходи, як правило, використовують механізацію вимірювання інерціальної вимірювальної установки як крок передбачення або припущення руху транспортного засобу (сталий швидкісний рух, сталый прискорений рух тощо). Щодо моделі спостереження, то за умови візуального *SLAM* вона часто ґрунтується на моделі зворотної глибини або на класичній перспективній моделі огляду. Щодо *LiDAR*-, *RGB-D*- або *RADAR*-підходів, то модель спостереження значно спрощується, оскільки спостереження є прямим *2D/3D*-вимірюванням *3D*-простору.

$$P2D = \Pi(P3D, K, T), \quad (8)$$

де  $T = [R, t]$  – жорстке перетворення, що забезпечує *6D*-позу датчика,  $K$  – внутрішні параметри датчика, а  $\Pi(\cdot)$  – функція перспективної проєкції. Як видно, цю функцію потрібно перевернути, щоб відповідати класичній моделі спостереження ( $P3D = g(P2D, K, T)$ ). Однак така інверсія не є простою, тому крок оцінювання часто відкладається до додаткового спостереження *P2D*. Потім виконується триангуляція  $P3D_{map}$ .

### Метод лазерного сканування (*LiDAR-SLAM*)

*LiDAR* може виявляти відстань до перешкод, і це є найкращим датчиком для побудови сіткової карти, що відображає структуру й перешкоди на площині руху робота. У ранніх дослідженнях *SLAM* часто використовували *LiDAR* як основний датчик. Розширений фільтр Калмана (*EKF*) застосовується для оцінювання положення робота, але продуктивність не була ідеальною. Для деяких сильно нелінійних систем цей метод спричинить більшу кількість помилок відрізання, унаслідок чого матимемо неточне позиціонування та картографування. Були запропоновані підходи, основані на фільтрах частинок, оскільки вони ефективно уникають нелінійних проблем, але це також призводить до проблеми зі збільшенням обчислювальної складності та зі зростанням кількості частинок. 2007 р. *Grisetti* запропонував важливий метод *LiDAR-SLAM*, названий *Gmapping*. Він ґрунтується на *Rao-Blackwellized*-фільтрах частинок (*RBPF*), який збільшує точність позиціонування та зменшує обчислювальну складність шляхом покращення запропонованого розподілу та адаптивної техніки перевірки [10].

Як альтернатива ймовірнісним підходам, останніми роками популярні оптимізаційні методи. 2010 р. *Kurt Konolige* запропонував представницький метод *Karto-SLAM*, що використовує розріджене коригування положення, щоб вирішити проблему безпосереднього матричного розв'язання в нелінійній оптимізації. *Hector SLAM* [10], запропонований 2011 р., оцінює застосування методу Гаусса–Ньютона для вирішення проблеми сканування відповідності. Цей метод не потребує інформації відометра, але вимагає високоточного *LiDAR*. 2016 р. *Google* запропонувала метод під назвою *Cartographer*, застосувавши лазерне замкнення петлі до підмап та глобальної карти. Завдяки цьому методу накопичувана похибка зменшується [5].

### Візуальна одометрія (*Visual-SLAM*)

Використання візуальних сенсорів для побудови карти середовища є ще одним актуальним напрямом для навігації роботів. На відміну від *LiDAR-SLAM*, *Visual-SLAM* є складнішим, оскільки зображення містить занадто багато інформації, але має труднощі у вимірюванні відстані. Оцінювання руху робота способом зіставлення виокремлених ознак

зображення за різними позами для побудови карти ознак є загальним методом для *Visual-SLAM*.

*Mono-SLAM*, запропонований 2007 р., вважається походженням багатьох *Visual-SLAM*. Розширений фільтр Калмана (*EKF*) використовується як зворотний елемент для відстеження розріджених ознак у передній частині. Невизначеність виражена функцією щільності ймовірності. З моделі спостереження та рекурсивних обчислень отримують середнє та дисперсію апостеріорного розподілу ймовірності. У роботі використовується *RBPF* для реалізації *Visual-SLAM*. Цей метод уникнув проблеми нелінійності та має високу точність, але потребує значної кількості частинок, що підвищує обчислювальну складність. *PTAM* є репрезентативною роботою з *Visual-SLAM*, яка запропонувала простий і ефективний метод виділення ключових кадрів, а також паралельну структуру реального часу для відстеження та зворотної нелінійної оптимізації картографування. Уперше запропоновано ідею розділення передньої та задньої частини, що призвело до структурного проєктування багатьох методів *SLAM*.

*ORB-SLAM* [10] вважається віхою візуальної *SLAM*. Застосувавши функції *Oriented FAST* і *Rotated BRIEF* (*ORB*) та модель "мішок слів" (*BOW*), цей метод може створити в реальному часі стабільну карту особливостей навколишнього середовища в багатьох ситуаціях. Виявлення петель та їх закриття через *BOW* є вагомим внеском цієї роботи, адже ефективно запобігає кумулятивній помилці та може бути швидко відновлене після втрати відстеження.

Останнім часом, на відміну від методів на основі особливостей, досліджувалися директивні методи візуальної *SLAM* шляхом оцінювання руху робота через значення пікселів безпосередньо. Густинне вирівнювання зображень на основі кожного пікселя зображень, запропоноване в джерелі, може побудувати густу 3D-карту оточення. У процесі роботи з джерелом було побудовано напівгусту карту способом оцінювання значень глибини пікселів із великим градієнтом на зображенні. *Engel* тощо запропонували алгоритм *LSD-SLAM* (*Large-Scale Direct Monocular SLAM*). Ядро цього алгоритму полягає в застосуванні безпосереднього методу до напівгустого монокулярного *SLAM*, що мало бути попередньо. 2014 р. *Forster* та інші запропонували *SVO* (*Semi-Direct Monocular Visual Odometry*), що називається "розрідженим прямим методом", який поєднує ключові точки особливостей з прямими

методами для відстеження деяких ключових точок (наприклад, кутів), а потім оцінює рух камери та її положення відповідно до інформації навколо ключових точок. Цей метод працює швидко для безпілотних літальних апаратів (UAV), додавши спеціальні обмеження та оптимізацію до таких застосувань.

Камера *RGB-D* може надавати інформацію як про колір, так і про глибину в полі зору. Вона є найбільш ефективним датчиком для створення повної *3D*-карти сцени. У роботі пропонується метод *Kinect fusion*, що використовує зображення глибини, отримані з *Kinect*, для вимірювання мінімальної відстані кожного пікселя на кожному кадрі та об'єднує всі зображення глибини для глобальної карти інформації. Крім того, конструюється функція помилки за допомогою фотометричної та геометричної інформації пікселів зображення. Положення камери отримується шляхом мінімізації функції помилки. Проблема картографування розглядається як подання графа положення. Робота є кращим безпосереднім методом *RGB-D SLAM*. Цей метод поєднує помилку інтенсивності та помилку глибини пікселів як функції помилки й мінімізує функцію витрат, щоб отримати оптимальне положення камери. Цей процес реалізовується за допомогою *g2o*. Запропоновано метод вилучення ключових кадрів на основі ентропії та метод виявлення замкненого контура, що значно зменшує помилку маршруту [5].

### Об'єднання інформації з декількох сенсорів

Введення даних допоміжних датчиків може покращити стійкість системи *SLAM*. На сьогодні для *LiDAR-SLAM* і *Visual-SLAM* найбільш часто використовуються допоміжні сенсори – енкодер та інерціальна вимірювальна одиниця (*IBO*), які можуть надавати додаткову інформацію про рух робота. Системи *SLAM* з такими допоміжними сенсорами зазвичай працюють краще.

Останнім часом на основі робіт *LiDAR-SLAM* та *Visual-SLAM* деякі вчені почали досліджувати інтеграції цих двох основних сенсорів [11–15]. У праці [11] автори застосували візуальний одометр з метою надання початкових значень для двовимірного лазерного методу ітеративного замикання точок (*ICP*) на невеликому БПЛА та досягли хороших результатів у реальному часі та точності. У дослідженні [12] запропоновано графову структуру на основі *SLAM* з монокулярною камерою та лазером із припущенням, що стіна

перпендикулярна до землі та вертикально плоска. У роботі [13] інтегруються різні передові методи *SLAM* на основі зору, лазера та інерціальних вимірювань за допомогою розширеного фільтра Калмана для БПЛА в приміщенні. У дослідженні [14] представлено метод локалізації, оснований на співпраці між повітряними й наземними роботами в закритому приміщенні, де сенсор *RGB-D* та *2D-LiDAR* прикріплені до БПЛА для створення *2,5D*-карти висот і його локації. Метод оцінювання масштабу та корекції дрейфу способом поєднання монолазерного далекоміра й камери для моно-*SLAM* описано в праці [15]. У роботі [16] запропоновано візуальну *SLAM*-систему, що поєднує зображення, отримані з камери, і розріджену глибинну інформацію, отриману з *3D-LiDAR*, за допомогою прямого методу. У дослідженні [17] виконано фузію *EKF* на позиції, обчислені модулем *LiDAR* та модулем зору, і запропоновано покращену стратегію для відстеження візуальної *SLAM*, коли втрачається об'єкт. З камерою та *LiDAR* стають стандартними конфігурації для роботів, а фузія лазерної та візуальної інформації стане нагальною темою досліджень для *SLAM*, оскільки вона може забезпечити більш стійкий результат для реального застосування [5].

### Проблематика застосування

Загалом, методи *LiDAR-SLAM* будують карту зайнятості сітки, яка готова для планування маршруту та керування навігацією. Однак для будівництва більших карт потрібне виявлення та корекція замкненого циклу, що не є легким завданням для сіткової карти. Оскільки отримані способом сканування дані є двовимірними хмарами точок, що не мають явних ознак і дуже схожі між собою, виявлення замкненого циклу на основі результатів сканування безпосередньо часто є неефективним. І цей недолік також поширюється на швидку функцію переміщення, коли робот працює із заданою картою. У навігаційному пакеті, що надається операційною системою *ROS*, робот потребує ручного введення початкової позиції перед автоматичною навігацією та рухом.

З іншого боку, більшість підходів *Visual-SLAM* створюють карти ознак, що добре підходять для локалізації, але не ефективні для планування маршруту. *RGB-D* або *3D-LiDAR* здатні побудувати повну *3D*-сцену, але це обмежено використанням через високу вартість обчислення або розміщення.

Для споживчих роботів вартість датчиків та оброблювального обладнання є чутливою. Маломіцні датчики *LiDAR* стають популярними. Однак створення надійної системи навігації низької вартості не є легкою роботою. Оскільки маломіцні датчики *LiDAR* мають значно гірші характеристики щодо частоти, роздільної здатності та точності, ніж звичайні датчики. У багатьох працях, наприклад [18], уже запропоновані методи для покращення точності зведення сканування для маломіцних датчиків *LiDAR*, проте це ефективно лише для сусідніх позицій. Акумулявання похибок може швидко наростати та призводити до невдачі в будівництві більших карт. Знайти ефективне та надійне рішення *SLAM* і переміщення з низькою вартістю обчислення та розміщення все ще є викликом для комерційно застосовуваних сервісних роботів [5].

### Метод співпраці датчиків 2D LRF та RGB-D

У статті пропонується метод локалізації інтелектуального робота за допомогою співпраці датчиків лазерного сканування (2D LRF) та глибинного відтворення зображень (RGB-D) для побудови 2,5D-мапи картографування.

У побудові більш складних репрезентацій, ніж 2D, для забезпечення правильного подання навколишнього середовища, використовуються 3D-репрезентації та 3D-сенсори.

*SLAM* 2,5D-картографування середовища – це метод, який застосовується в системах *SLAM*, де карта середовища подана у вигляді 2,5D-карти висоти.

2,5D-карта висоти – це вид картографічної інформації, що містить висотні дані про територію з деяким рівнем деталізації. Вона є двовимірним зображенням, де кожний піксель містить інформацію про висоту точки над поверхнею землі. На відміну від 3D, карти 2,5D відтворюють тільки вертикальні висотні дані, а не повні 3D-моделі.

### Подання 2,5D-картографування

Це подання ґрунтується на двовимірній сітці квадратних комірок ( $C_{ij}, i, j \in \mathbb{N}$ ). Кожна точка в тривимірній хмарі точок перетворюється в систему координат робота та проектується на площину  $XY$ , накладаючись на комірку сітки  $C_{ij}$ . Локальна сіткова мапа має розмір  $N_{rows} \times N_{columns}$ , роздільна

здатність  $s_{cell}$  на комірку та координати центра мапи ( $p_m^o = x_m^o, y_m^o$ ). Перетворення між декартовими координатами  $(x, y)$  та сітковими координатами  $(i, j)$  задається таким чином:

$$(i, j) \leftarrow \left( \frac{x - x_m^o + \frac{s_{cell}}{2}}{s_{cell}} + \frac{N_{cols}}{2}, \frac{y - y_m^o + \frac{s_{cell}}{2}}{s_{cell}} + \frac{N_{rows}}{2} \right). \quad (9)$$

Перетворення координат сітки  $(i, j)$  у декартові координати  $(x, y)$  задається так:

$$(x, y) \leftarrow \left( s_{cell} \left( i - \frac{N_{cols}}{2} \right) + x_m^o, s_{cell} \left( j - \frac{N_{rows}}{2} \right) + y_m^o \right). \quad (10)$$

Кожна комірка в сітці містить два можливі значення: порожнє, якщо в цій комірці немає точки, або з найменшим і найбільшим значеннями  $z$ -координат точок, які належать до цієї комірки. Висотний воксель отримується на основі значень висоти комірок (приклад зображено на рис. 3).

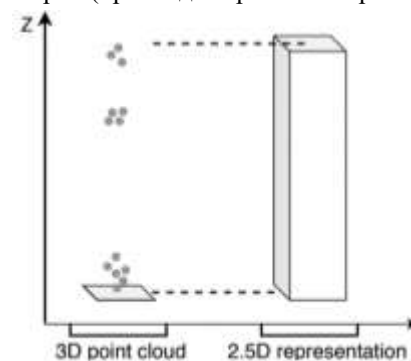


Рис. 3. 2,5D-подання [19]

Для отримання координат локальної карти в глобальних координатах застосовується жорстке перетворення. Для цього, початкова глобальна 2D-карта конвертується у 2,5D-карту, де кожній комірці призначається мінімальна й максимальна висота. Ця мапа оновлюється висотними вокселями, отриманими з локальної карти. Було розроблено алгоритм трасування 3D-ліній для висотних вокселів, щоб обчислити вільний простір між датчиком і кожним висотним вокселем. Координати датчика й кожного висотного вокселю проектується на площину 2D-сітки карти. Комірki на лінії з 3D-датчика до висотного вокселю обчислюються за допомогою алгоритму Брезенхема [20]. Для кожного висотного вокселю відомі мінімальні та максимальні значення висоти, та визначаються дві лінійні моделі від датчика, що надають вільний висотний воксель для кожної комірки (рис. 4).

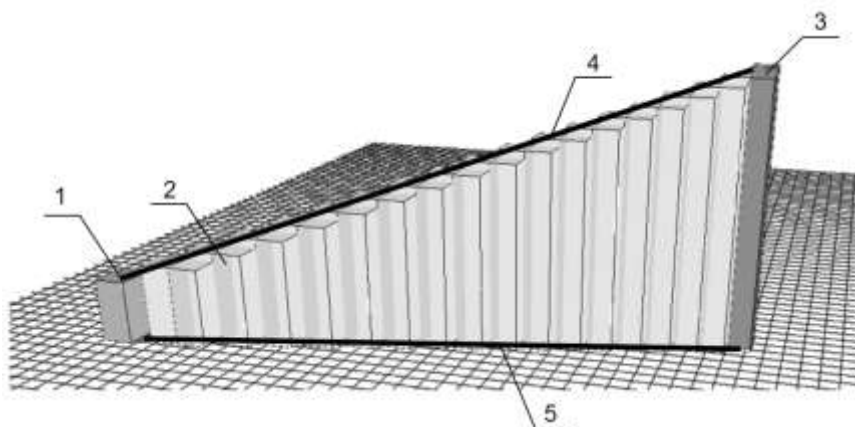


Рис. 4. Трасування лінії для одного вимірюваного висотного вокселю [19]

Перший датчик зображений під номером 1, вільні висотні вокселі – під номером 2, вимірюваний висотний воксель позначений номером 3, а лінії, що з'єднують максимальне та мінімальне значення  $z$ , – номерами 4 та 5 відповідно.

Подання, що генерується з 3D-хмари точок, надає оновлення карти, заданої заздалегідь, змінюючи мінімальні та максимальні значення висоти кожної комірки за допомогою згаданого алгоритму. Усі комірки, ідентифіковані алгоритмом Брезенхема, які містять дані, перекриваються з відповідним вільним висотним вокселем. Розглядаючи мінімальні та максимальні значення висоти висотного вокселю ( $h_{min}, h_{max}$ ) та вільного висотного вокселю ( $fh_{min}, fh_{max}$ ), маємо такі спостереження відповідно до перелічених нижче правил.

1. Якщо вільний висотний воксель перекривається знизу висотного вокселю, то мінімальне значення висоти оновлюється до  $fh_{max}$  (рис. 5, a).

2. Якщо вільний висотний воксель перекривається зверху висотного вокселю, то максимальне значення висоти оновлюється до  $fh_{min}$  (рис. 5, b).

3. Якщо жоден вільний висотний воксель не перекривається з висотним вокселем, то змін не відбувається (рис. 5, c).

4. Якщо вільний висотний воксель повністю перекриває висотний воксель, то інформація вилучається (рис. 5, d).

5. Якщо висотний воксель з вільною висотою є частиною висотного вокселю, то використовується індекс Жаккара, також відомий як "перетин-об'єднання" ( $IoU$ ). Оцінка визначається як відношення площі перетину до площі об'єднання. Якщо оцінка більша за 0,5, то вилучаються дані, в іншому разі жодних змін не відбувається (рис. 5, e).

Висотний воксель зображено білим, а вільний висотний воксель з вільною висотою – світло-сірим кольором.

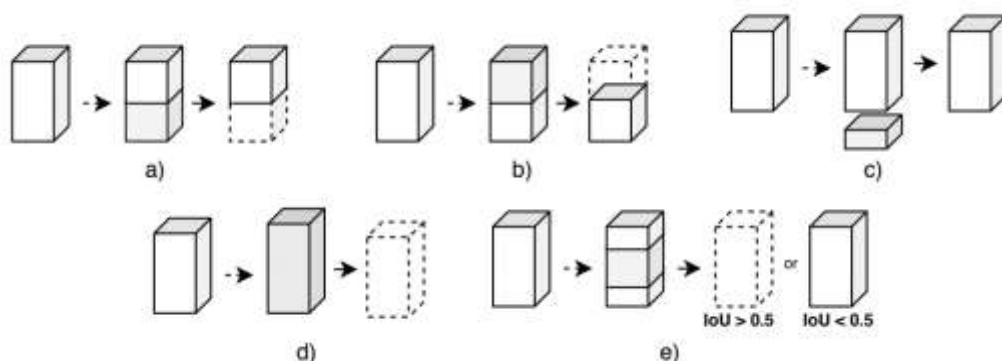


Рис. 5. Ілюстрація методу перекриття для оновлення 2,5D-карти [19]

### Картографування та локалізація у 2,5D

У цьому підході фільтр на основі  $KLD$  оцінює позу робота, використовуючи результати одометрії,

апріорну 2D-карту та 3D-хмару точок, і генерує оновлену 2,5D-карту. Маємо 3D-хмару точок, що обробляється для отримання локального подання у вигляді 2,5D. Спочатку задана апріорна 2D-карта,

яка перетворюється на 2,5D-карту, як описано вище. Карта 2,5D регулярно оновлюється за допомогою позиції, оціненої фільтром, та локального подання. Глобальна й локальна карти також описані вище.

Як показано на рис. 6, обчислення ваги частинок ґрунтується на апіорній 2,5D-карті, локальній та глобальній карті.

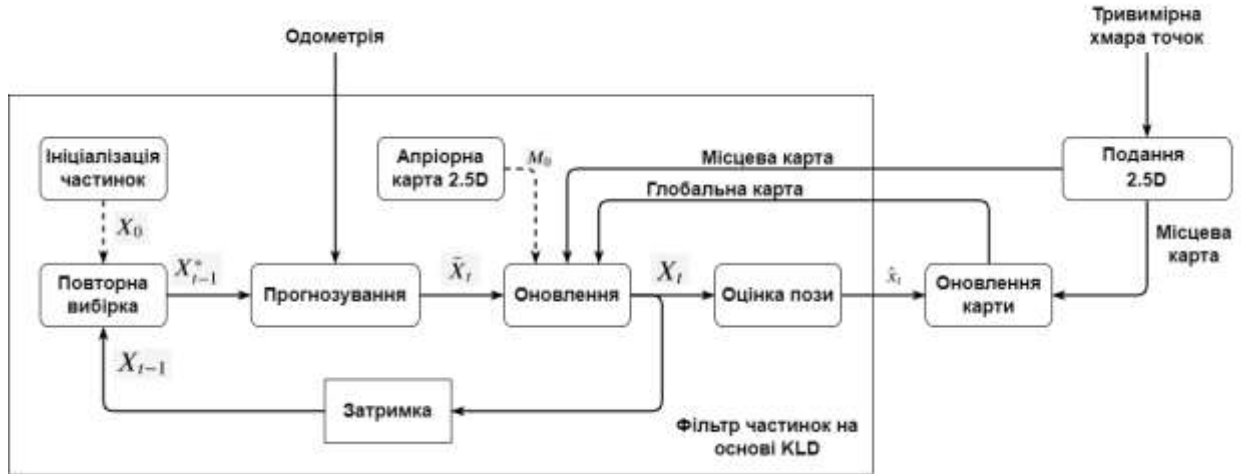


Рис. 6. Діаграма фільтра частинок на основі KLD

На діаграмі фільтра частинок на основі KLD  $X_0$  – початковий набір частинок;  $X_{t-1}$  та  $X_t$  – попередній та поточний набори частинок;  $X_{t-1}^*$  – переставлений попередній набір частинок;  $\bar{X}_t$  – фактичний передбачений набір;  $\hat{x}_t$  – фактично оцінена поза;  $M_0$  – початкова 2,5D-карта; *LocalMap* та *GlobalMap* – локальна та глобальна карти, отримані з модуля картографування.

Крок оновлення фільтра на основі KLD описаний в алгоритмі етапу оновлення (рис. 7) і має на вході частинку від даних із локальної та глобальної карти. Обчислення ваги частинки виконується в рядках 2–7, та деякі вокселі локальної 2,5D-карти пропускаються (ігноруючи

$C$  вокселів для всіх оброблених вокселів), щоб зменшити час оброблення. Кожен обраний воксель перетворюється у світову систему координат, застосовуючи жорстку трансформацію (рядок 3) і потім використовується в процедурі пошуку найближчого сусіднього вокселю. Ця процедура шукає в глобальній 2,5D-карті найближчий воксель до обраного вокселю (з локальної 2,5D-карти) та повертає координати вокселю, що має найвищий перетин (IoU) з обраним вокселем. У рядку 5 вага обчислюється шляхом різниці між обраним вокселем і вокселем (отриманим із процедури найближчого сусіда) в нормальних розподілах  $N_x$  та  $N_y$ .

```

Data: Particle  $x_t^{[n]} = \{x_t^{[n]}, y_t^{[n]}, \theta_t^{[n]}\}$ , set of voxels from the local map (LM), set of voxels from the global map (GM)
and  $C$  a constant.
1  $w_t^{[n]} \leftarrow 0, j \leftarrow 0;$ 
2 for  $i = 1; i < \text{size}(\text{LM}); i = i + C$  do
3    $p_t^i \leftarrow \text{transform}(x_t^{[n]}, LM_x^i, LM_y^i, LM_{zmin}^i, LM_{zmax}^i);$  // Transforms the selected voxel to the world
   frame
4    $(x_z^i, y_z^i) \leftarrow \text{nearest}(p_t^i, GM);$  // Checks neighborhood
5    $w_t^{[n]} \leftarrow w_t^{[n]} + \mathcal{N}_x(0, p_t^i - x_z^i) \cdot \mathcal{N}_y(0, p_t^i - y_z^i);$ 
6    $j \leftarrow j + 1;$ 
7 end
8  $w_t^{[n]} \leftarrow \frac{w_t^{[n]}}{j};$ 
9 return  $w_t^{[n]}$ 
    
```

Рис. 7. Алгоритм етапу оновлення [19]

## Результати досліджень

Для проведення моделювання було прийнято рішення використовувати операційну систему для роботів ROS, яка пропонує різноманітні способи для розв'язання проблем SLAM [21].

ROS має два основні складники: операційна система *ros*, описана вище, і набір пакетів *ros-pkg*, що підтримуються користувачами та організовані в набори, які називаються стеками. Ці пакети реалізують різні функції робототехніки, такі як SLAM, планування, сприйняття, моделювання тощо [22].

Для візуалізації інтелектуальних роботів і навколишнього середовища були використані пакети *Gazebo* та *Rviz*, які є частиною пакету *Robot Operating System*. *Gazebo* дає змогу моделювати фізичні властивості інтелектуального робота, довкілля, показники різних датчиків тощо. З допомогою *Rviz* можна створювати віртуальну модель робота та візуалізувати карти, маршрути, інформацію датчиків.

Різницю між ними можна підсумувати цитатою з книги "Програмування роботів з ROS" М. Квіглі – одного з розробників ROS: "*Rviz* демонструє, що відбувається на думку робота, а *Gazebo* показує, що відбувається насправді".

Сцена тестового приміщення (рис. 8–9) створена за допомогою інструментів *Gazebo (Gzweb)* та пакетів *AWS RoboMaker*.

*AWS RoboMaker* – хмарний сервіс імітаційного моделювання, що надає розробникам роботів змогу запускати, масштабувати й автоматизувати імітаційне моделювання без необхідності керувати якоюсь інфраструктурою. *Gzweb* – це клієнт *WebGL* для *Gazebo*.

Планування руху мобільного робота є складним завданням, що має враховувати не тільки об'єкти, відображені на карті, а й ті, які відсутні на ній під час побудови. Наприклад, якщо на шляху з'являється новий об'єкт, потрібно розпізнати його та перебудувати маршрут. Для вирішення цього

завдання ROS надає набір пакетів, що називається навігаційним стеком, який об'єднує компоненти *move\_base* та *amcl*.

Навігаційний стек ROS – двовимірний. Він отримує інформацію про позицію мобільної бази, одометрії та потоків датчиків і виводить команди безпечної швидкості для мобільної бази. Для побудови маршруту навігаційний стек використовує набір пакетів, зокрема *map\_server* для 2D-карт, *amcl* для локалізації на мапі, повідомлення датчиків та одометрії, а також *move\_base* для об'єднання всіх повідомлень і виведення команди швидкості. Навігаційний стек зберігає інформацію про перешкоди в середовищі у двох картах витрат. Глобальна карта витрат використовується для довгострокового планування всього середовища, тоді як локальна карта витрат застосовується для нетривалого планування й уникнення перешкод.

Під час симуляції моделювання навколишнього середовища (рис. 10, 11) отримано й порівняно карту 2D-LIDAR (рис. 12) та 2,5D-карту висот (рис. 13).

Щоб скласти початкову карту довкілля, було записано серію точок шляху. Ці маршрутні точки маневрували мобільною платформою з її початкового положення за межами простору, у кількох кімнатах і навколо них, і поверталися назад у вихідне положення. Це привело до повторюваного шляху з достатнім покриттям датчика з метою підтримки розроблення карти середовища. Щоб створити карту сітки зайнятості, яка застосовуватиметься для майбутнього планування шляху, вихід датчика *Kinect* був змінений, щоб він виглядав як вихідний результат лазерного сканування. Датчик *Kinect* надає інформацію про глибину в тривимірній хмарі точок. Однак для 2D-картографування все, що необхідно, – це серія зображень глибини на одній висоті. Це зменшує витрати на обчислення та значно спрощує навігацію в базових середовищах.



Рис. 8. Сцена тестового приміщення (вигляд спереду)

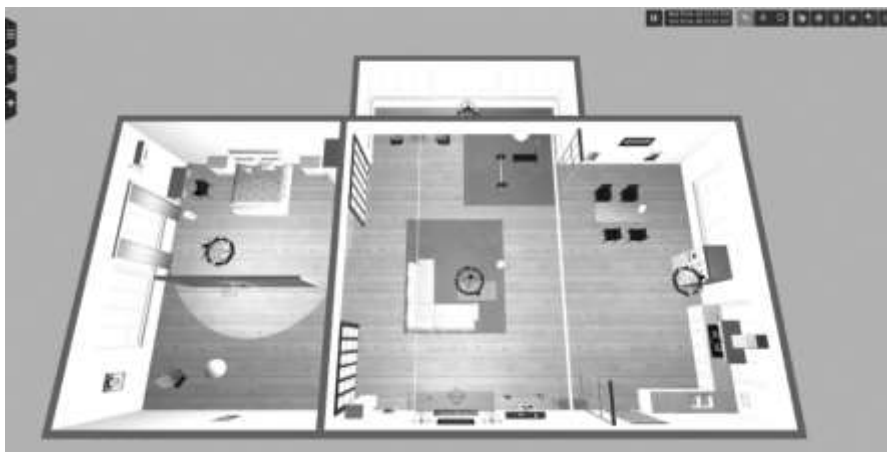


Рис. 9. Сцена тестового приміщення (вигляд зверху)

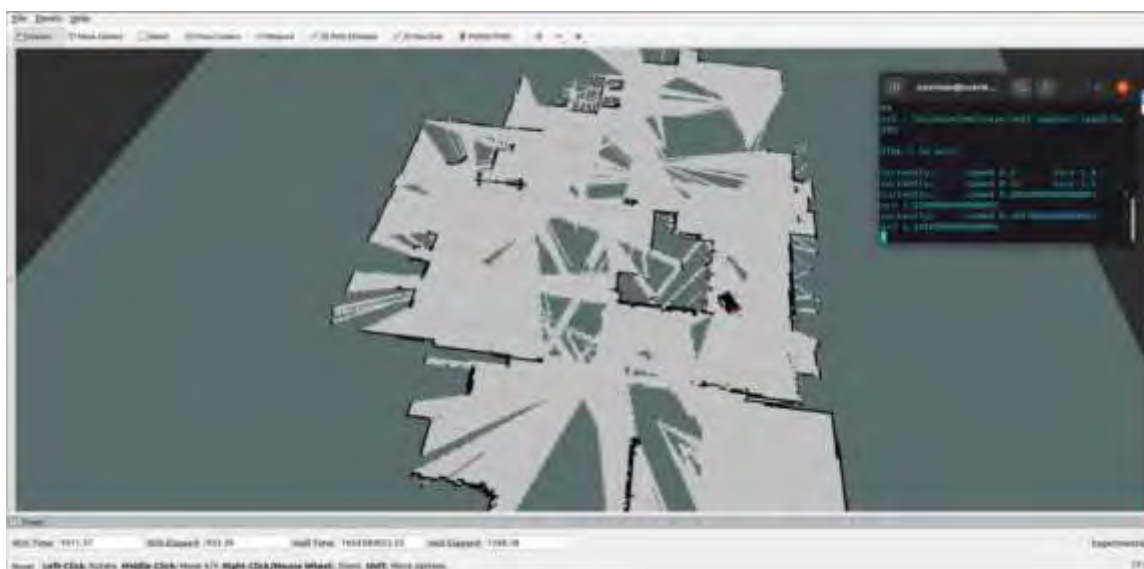


Рис. 10. Процес симуляції моделювання 2D-картографування

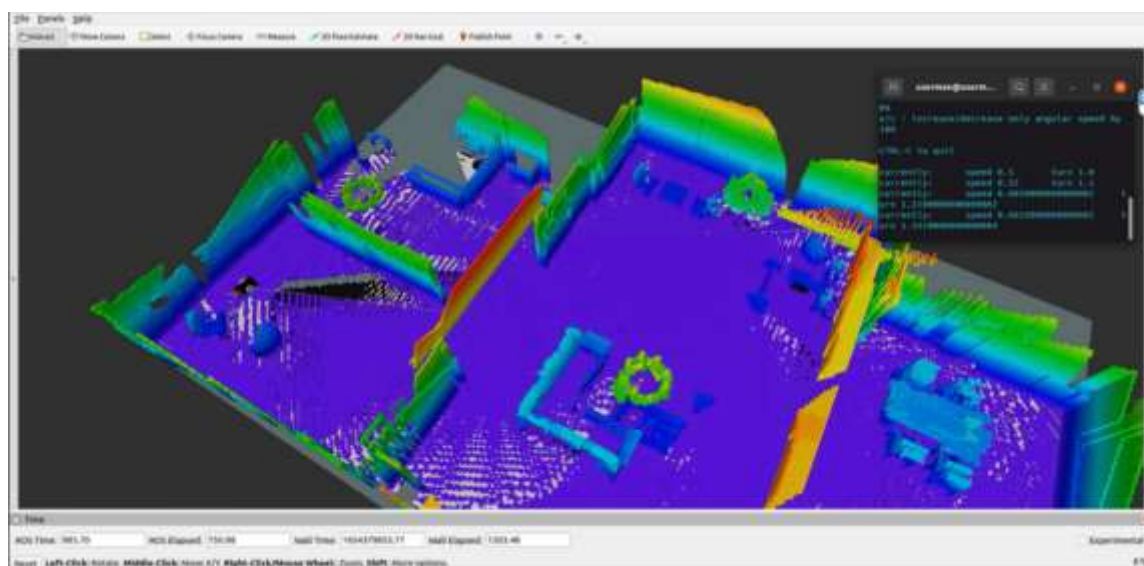


Рис. 11. Процес симуляції моделювання 2,5D-картографування

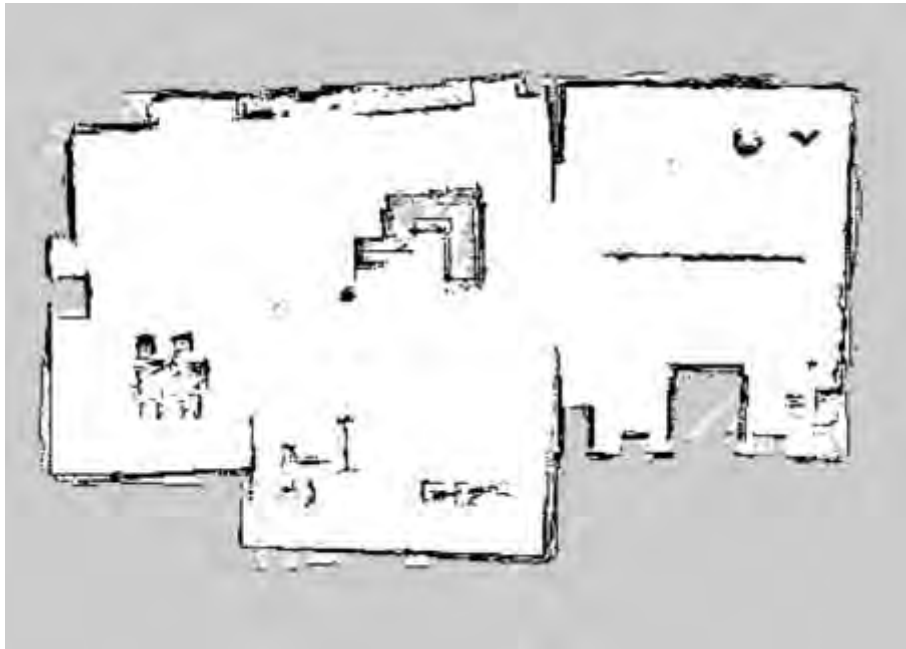


Рис. 12. Отримане 2D-просторове середовище

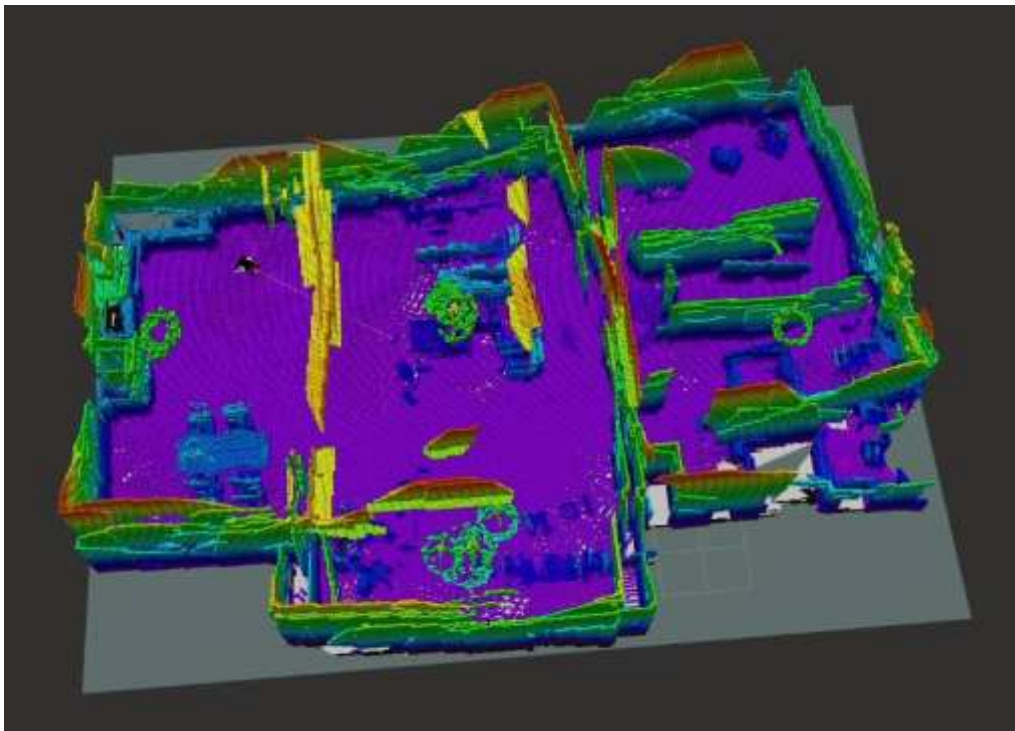


Рис. 13. Отримана 2,5D-карта висот

Навігаційний стек зберігає інформацію про перешкоди у світі у двох картах витрат. Глобальна карта витрат використовується для довгострокового планування всього середовища, тоді як локальна карта витрат – для нетривалого планування й уникнення перешкод. Один файл конфігурації містить параметри, спільні для обох карт витрат,

тоді як інші файли конфігурації мають параметри, властиві для глобальної або локальної карти витрат. Це створює локальний йі глобальний план, що можна візуалізувати. Крім того, і локальні, і глобальні карти витрат можна унаочнити, щоб відобразити перешкоди, виявлені в навколишньому середовищі (рис. 14).

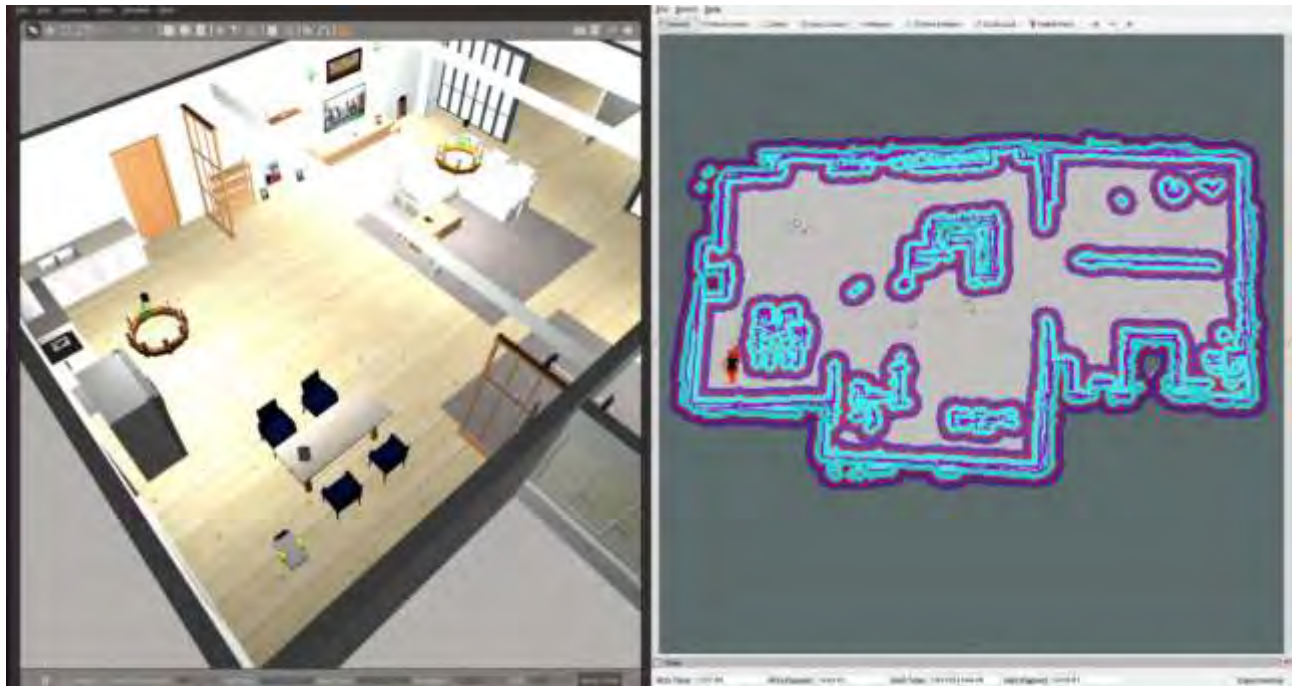


Рис. 14. Візуалізація карт вартості планування шляху

На рис. 14 можна бачити 3D-модель приміщення (ліворуч) та приклад результату автономної навігації інтелектуального робота з використанням раніше згенерованої 2,5D-карти.

#### Висновки й перспективи подальшого дослідження

У статті подано результати моделювання алгоритму методу локалізації інтелектуального робота за допомогою співпраці датчиків лазерного сканування (2D LRF) та глибинного відтворення зображень (RGB-D) для побудови 2,5D-карти довкілля інтелектуального робота. Це дало змогу провести симуляцію роботи методу 2,5D-картографування та визначити місце розташування й орієнтацію мобільного робота в просторі в режимі реального часу за допомогою пакетів ROS.

Наразі 2D-сканування не здатне виявити певні перешкоди через їх висоту, тоді як 3D-сканування містить великі обсяги даних, що потрібно обчислити для картографування. Це ускладнює використання алгоритмів у режимі реального часу. Щоб вирішити ці проблеми, у статті запропоновано 2,5D-метод відображення висот.

Порівняно методи 2D/3D- та 2,5D-картографування й зроблено конкретні висновки.

Підхід 2,5D-картографування та локалізації є меншим обчислювальним бар'єром завдяки простоті його подання. Результати, здобуті для 2,5D-картографування та локалізації, показують, що цей підхід якісно відтворює довкілля та успішно локалізує робота на карті 2,5D. Оцінено позиції робота за умови використання лазерного сканування та 3D-хмари точок як вхідних даних. Крім того, результати є послідовними в процесі розгляду тестів з оновленням карти 2,5D та без оновлення.

Здобуті результати є обнадійливими й демонструють потенціал роботи об'єднаних методів SLAM, що співпрацюють для забезпечення точного виконання одночасної локалізації та картографування інтелектуальних роботів у режимі реального часу.

Такий підхід розширює технології, не замінюючи наявні робочі пропозиції, і дає змогу використовувати сучасні методи для всебічного виявлення та розпізнавання довкілля з допомогою ефективного локалізаційного й картографічного підходу, надаючи більш точні результати з використанням менших ресурсів. У подальших дослідженнях плануємо вдосконалити запропонований метод для навігації автономних інтелектуальних роботів у процесі використання на пересічній місцевості з урахуванням перепадів висот.

**Список літератури**

1. Nevliudov I., Novoselov S., Sychova O., Mospan D. Multithreaded Software Control of Industrial Manipulator Movement, *IEEE 4th International Conference on Modern Electrical and Energy System (MEES)*. Kremenchuk, Ukraine. 2022. P. 1–6. DOI: 10.1109/MEES58014.2022.10005675
2. Сухачов К. Сучасні методи одночасної локалізації і картографування в режимі реального часу. *Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку*. Матеріали Всеукраїнської науково-практичної Інтернет-конференції. Черкаси, 2023. С. 77–79. URL: <https://conference.ikto.net/>
3. Nevliudov I., Novoselov S., Sychova O., Tesliuk S. Development of the Architecture of the Base Platform Agricultural Robot for Determining the Trajectory Using the Method of Visual Odometry. *IEEE XVII-th International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)*. Polyana (Zakarpattia). Ukraine. 2021. P. 64–68. DOI: 10.1109/MEMSTECH53091.2021.9468008
4. Khan M. S. A. et al. Investigation of Widely Used SLAM Sensors Using Analytical Hierarchy Process. *Journal of sensors*. 2022. Vol. 2022. P. 1–15. DOI: <https://doi.org/10.1155/2022/5428097>
5. Jiang G. et al. A simultaneous localization and mapping (SLAM) framework for 2.5D map building based on low-cost lidar and vision fusion. *Applied sciences*. 2019. Vol. 9, № 10. 2105. DOI: <https://doi.org/10.3390/app9102105>
6. Rosas-Cervantes A. et al. Sensors Multi-Robot 2.5 D localization and mapping using a monte carlo algorithm on a multi-level surface. *208 MDPI Journals Awarded Impact Factor*. 2021. Vol. 21. № 13. 4588. DOI: <https://doi.org/10.3390/s21134588>
7. Debeunne C., Vivet D. A review of visual-lidar fusion based simultaneous localization and mapping. *Sensors*. 2020. Vol. 20. №. 7. 2068. DOI: <https://doi.org/10.3390/s20072068>
8. Bustos A. P. et al. Visual SLAM: why bundle adjust? *International conference on robotics and automation (ICRA)*. Montreal, QC, Canada. 20–24 May. 2019. 1043. DOI: <https://doi.org/10.1109/icra.2019.8793749>
9. O'Mahony N. et al. Deep learning for visual navigation of unmanned ground vehicles: a review. *29th Irish signals and systems conference (ISSC)*. Belfast. 21–22 June 2018. 859. DOI: <https://doi.org/10.1109/issc.2018.8585381>
10. Freitas C. M. Autonomous navigation with simultaneous localization and mapping in/outdoor: master's thesis. 2020. URL: <https://hdl.handle.net/10216/128968>
11. Lynen S. et al. A robust and modular multi-sensor fusion approach applied to MAV navigation. *IEEE/RSJ international conference on intelligent robots and systems (IROS 2013)*. Tokyo. 3–7 November. 2013. 6290. DOI: <https://doi.org/10.1109/iros.2013.6696917>
12. Oh T. et al. Graph structure-based simultaneous localization and mapping using a hybrid method of 2D laser scan and monocular camera image in environments with laser scan ambiguity. *Sensors*. 2015. Vol. 15. № 7. P. 15830–15852. DOI: <https://doi.org/10.3390/s150715830>
13. López E. et al. A multi-sensorial simultaneous localization and mapping (SLAM) system for low-cost micro aerial vehicles in gps-denied environments. *Sensors*. 2017. Vol. 17. № 4. 802. DOI: <https://doi.org/10.3390/s17040802>
14. Nam T., Shim J., Cho Y. A 2.5D map-based mobile robot localization via cooperation of aerial and ground robots. *Sensors*. 2017. Vol. 17. №. 12. 2730. DOI: <https://doi.org/10.3390/s17122730>
15. Zhang Z. et al. Scale estimation and correction of the monocular simultaneous localization and mapping (SLAM) based on fusion of 1D laser range finder and vision data. *Sensors*. 2018. Vol. 18. №. 6. 1948. DOI: <https://doi.org/10.3390/s18061948>
16. Shin Y.-S., Park Y. S., Kim A. Direct visual SLAM using sparse depth for camera-lidar system. *IEEE international conference on robotics and automation (ICRA)*. Brisbane, QLD, Australia. 21–25 May 2018. P. 5144–5151. DOI: <https://doi.org/10.1109/icra.2018.8461102>
17. Xu Y., Ou Y., Xu T. SLAM of robot based on the fusion of vision and LIDAR. *IEEE international conference on cyborg and bionic systems (CBS)*. Shenzhen. 25–27 October 2018. 2058. DOI: <https://doi.org/10.1109/cbs.2018.8612212>
18. Jiang, G. et al. FFT-Based Scan-Matching for SLAM Applications with Low-Cost Laser Range Finders. *Applied sciences*. 2018. Vol. 9. № 1. 41. DOI: <https://doi.org/10.3390/app9010041>
19. Alves S. A. F. T. Particle-Filter based 3D mapping, localization and SLAM for indoor mobile robot navigation: master's thesis. 2019. URL: <http://hdl.handle.net/10316/87952>
20. Bresenham J. E. Algorithm for computer control of a digital plotter. *IBM Systems Journal*. 1965. Vol. 4. № 1. P. 25–30. DOI: <https://doi.org/10.1147/sj.41.0025>
21. Joseph L. Learning Robotics using Python: Design, simulate, program, and prototype an autonomous mobile robot using ROS, OpenCV, PCL, and Python, 2nd Edition. Packt Publishing, 2018. 280 p.

22. Nevliudov I., Sychova O., Reznichenko O., Novoselov S., Mospan D., Mospan V. Control System for Agricultural Robot Based on ROS. *IEEE International Conference on Modern Electrical and Energy Systems (MEES)*. Kremenchuk, Ukraine. 2021. P. 1–6. DOI: 10.1109/MEES52427.2021.9598560

## References

1. Nevliudov, I., Novoselov, S., Sychova, O., Mospan, D. (2022), "Multithreaded Software Control of Industrial Manipulator Movement", *IEEE 4th International Conference on Modern Electrical and Energy System (MEES)*, Kremenchuk, Ukraine. P. 1–6. DOI: 10.1109/MEES58014.2022.10005675
2. Sukhachov, K. (2023), "Modern methods of simultaneous localization and mapping in real time mode" ["Suchasni metody odnochasnoi lokalizatsii i kartohrafuvannia v rezhymi realnoho chasu"]. *Automation and computer-integrated technologies in production and education: status, achievements, development prospects*. All-Ukrainian scientific and practical Internet conference, Cherkasy, P. 77–79. available: <https://conference.ikto.net/>
3. Nevliudov, I., Novoselov, S., Sychova, O., Tesliuk, S. (2021), "Development of the Architecture of the Base Platform Agricultural Robot for Determining the Trajectory Using the Method of Visual Odometry", *IEEE XVII-th International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)*, Polyana (Zakarpattia), Ukraine, P. 64–68. DOI: 10.1109/MEMSTECH53091.2021.9468008
4. Khan, M. S. A. et al. (2022), "Investigation of Widely Used SLAM Sensors Using Analytical Hierarchy Process", *Journal of sensors*, Vol. 2022, P. 1–15. DOI: <https://doi.org/10.1155/2022/5428097>
5. Jiang, G. et al. (2019), "A simultaneous localization and mapping (SLAM) framework for 2.5D map building based on low-cost lidar and vision fusion", *Applied sciences*, Vol. 9, № 10, 2105. DOI: <https://doi.org/10.3390/app9102105>
6. Rosas-Cervantes, A. et al. (2021), "Sensors Multi-Robot 2.5 D localization and mapping using a monte carlo algorithm on a multi-level surface", *208 MDPI Journals Awarded Impact Factor*, Vol. 21. № 13. 4588. DOI: <https://doi.org/10.3390/s21134588>
7. Debeunne, C., Vivet, D. (2020), "A review of visual-lidar fusion based simultaneous localization and mapping", *Sensors*, Vol. 20, №. 7, 2068. DOI: <https://doi.org/10.3390/s20072068>
8. Bustos, A. P. et al. (2019), "Visual SLAM: why bundle adjust?", *International conference on robotics and automation (ICRA)*, Montreal, QC, Canada, 20–24 May, 1043. DOI: <https://doi.org/10.1109/icra.2019.8793749>
9. O'Mahony, N. et al. (2018), "Deep learning for visual navigation of unmanned ground vehicles: a review", *29th Irish signals and systems conference (ISSC)*, Belfast, 21–22 June, 859. DOI: <https://doi.org/10.1109/issc.2018.8585381>
10. Freitas, C. M. "Autonomous navigation with simultaneous localization and mapping in/outdoor: master's thesis", 2020, available at: <https://hdl.handle.net/10216/128968>
11. Lynen, S. et al. (2013), "A robust and modular multi-sensor fusion approach applied to MAV navigation", *IEEE/RSJ international conference on intelligent robots and systems (IROS 2013)*, Tokyo, 3–7 November, 6290. DOI: <https://doi.org/10.1109/iros.2013.6696917>
12. Oh, T. et al. (2015), "Graph structure-based simultaneous localization and mapping using a hybrid method of 2D laser scan and monocular camera image in environments with laser scan ambiguity", *Sensors*, Vol. 15, № 7, P. 15830–15852. DOI: <https://doi.org/10.3390/s150715830>
13. López, E. et al. (2017), "A multi-sensorial simultaneous localization and mapping (SLAM) system for low-cost micro aerial vehicles in gps-denied environments", *Sensors*, Vol. 17, № 4, 802. DOI: <https://doi.org/10.3390/s17040802>
14. Nam, T., Shim, J., Cho, Y. A. (2017), "2.5D map-based mobile robot localization via cooperation of aerial and ground robots", *Sensors*, Vol. 17, № 12, 2730. DOI: <https://doi.org/10.3390/s17122730>
15. Zhang, Z. et al. (2018), "Scale estimation and correction of the monocular simultaneous localization and mapping (SLAM) based on fusion of 1D laser range finder and vision data", *Sensors*, Vol. 18, № 6, 1948. DOI: <https://doi.org/10.3390/s18061948>
16. Shin, Y.-S., Park, Y. S., Kim, A. (2018), "Direct visual SLAM using sparse depth for camera-lidar system", *IEEE international conference on robotics and automation (ICRA)*, Brisbane, QLD, Australia, 21–25 May 2018, P. 5144–5151. DOI: <https://doi.org/10.1109/icra.2018.8461102>
17. Xu, Y., Ou, Y., Xu, T. (2018), "SLAM of robot based on the fusion of vision and LIDAR", *IEEE international conference on cyborg and bionic systems (CBS)*, Shenzhen, 25–27 October. 2058. DOI: <https://doi.org/10.1109/cbs.2018.8612212>
18. Jiang, G. et al. (2018), "FFT-Based Scan-Matching for SLAM Applications with Low-Cost Laser Range Finders", *Applied sciences*, Vol. 9, № 1. 41. DOI: <https://doi.org/10.3390/app9010041>
19. Alves, S. A. (2019), "Particle-Filter based 3D mapping, localization and SLAM for indoor mobile robot navigation: master's thesis", available at: <http://hdl.handle.net/10316/87952>
20. Bresenham, J. E. (1965), "Algorithm for computer control of a digital plotter", *IBM Systems Journal*, Vol. 4, № 1, P. 25–30. DOI: <https://doi.org/10.1147/sj.41.0025>
21. Joseph, L. (2018), *Learning Robotics using Python: Design, simulate, program, and prototype an autonomous mobile robot using ROS, OpenCV, PCL, and Python, 2nd Edition*, Packt Publishing, 280 p.
22. Nevliudov I., Sychova, O., Reznichenko, O., Novoselov, S., Mospan, D., Mospan, V. (2021), "Control System for Agricultural Robot Based on ROS", *IEEE International Conference on Modern Electrical and Energy Systems (MEES)*, Kremenchuk, Ukraine, P. 1–6. DOI: 10.1109/MEES52427.2021.9598560

*Відомості про авторів / About the Authors*

**Невлюдов Ігор Шакирович** – доктор технічних наук, професор, Харківський національний університет радіоелектроніки, завідувач кафедри комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки, Харків, Україна; e-mail: igor.nevlyudov@nure.ua; ORCID ID: <https://orcid.org/0000-0002-9837-2309>

**Новоселов Сергій Павлович** – кандидат технічних наук, доцент, Харківський національний університет радіоелектроніки, професор кафедри комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки, Харків, Україна; e-mail: sergiy.novoselov@nure.ua; ORCID ID: <https://orcid.org/0000-0002-3190-0592>

**Сухачов Костянтин Ігорович** – Харківський національний університет радіоелектроніки, магістр кафедри комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки, Харків, Україна; e-mail: kostiantyn.sukhachov@nure.ua; ORCID ID: <https://orcid.org/0009-0005-2277-9822>

**Nevlyudov Igor** – Doctor of Sciences (Engineering), Professor, Kharkiv National University of Radio Electronics, Head at the Department of Computer-Integrated Technologies, Automation and Mechatronics, Kharkiv, Ukraine.

**Novoselov Sergiy** – PhD (Engineering Sciences), Associate Professor, Kharkiv National University of Radio Electronics, Professor at the Department of Computer Integrated Technologies, Automation and Mechatronics, Kharkiv, Ukraine.

**Sukhachov Konstantin** – Kharkiv National University of Radio Electronics, Master's degree at the Department of Computer-Integrated Technologies, Automation, and Mechatronics, Kharkiv, Ukraine.

## METHOD OF SIMULTANEOUS LOCALIZATION AND MAPPING FOR CONSTRUCTION OF 2.5D MAPS OF THE ENVIRONMENT USING ROS

SLAM (Simultaneous Localization and Mapping) is a relevant topic of research and development in the field of robotics and computer vision. SLAM finds wide applications in various areas such as autonomous navigation of intelligent robots, solving problems in augmented and virtual reality, UAVs, and other systems. In recent years, SLAM has made significant progress due to the gradual development of its algorithms, the use of advanced sensors, and improvements in computational power of computers. **The subject** of this study is modern methods of real-time simultaneous localization and mapping. **The goal** of the research is to model the developed algorithm for constructing maps of the surrounding environment and determining the position and orientation of the intelligent robot in space in real-time using ROS packages. **The purpose** of this article is to demonstrate the results of combining SLAM methods and developing new approaches to solve simultaneous localization and mapping problems. In order to achieve the set objectives, a collaboration of laser scanning (2D LRF) and depth image reconstruction (RGB-D) **methods** was utilized for simultaneous localization and mapping of the intelligent robot and construction of a 2.5D environment map. The obtained **results** are promising and demonstrate the potential of the integrated SLAM methods, which collaborate to ensure accurate execution of simultaneous localization and mapping for intelligent robots in real-time mode. The proposed method allows for considering obstacle heights in constructing the map of the surrounding environment while requiring less computational power. **In conclusion**, this approach expands technologies without replacing existing working propositions and enables the use of modern methods for comprehensive detection and recognition of the surrounding environment through an efficient localization and mapping approach, providing more accurate results with fewer resources utilized.

**Keywords:** SLAM; ROS; 2D LRF; RGB-D; 2.5D height map; simultaneous localization and mapping methods; intelligent robot; position estimation; modeling and simulation.

*Бібліографічні описи / Bibliographic descriptions*

Невлюдов І. Ш., Новоселов С. П., Сухачов К. І. Метод одночасної локалізації та картографування для побудови 2,5D-карти навколишнього середовища засобами ROS. *Сучасний стан наукових досліджень та технологій в промисловості*. 2023. № 2 (24). С. 145–160. DOI: <https://doi.org/10.30837/ITSSI.2023.24.145>

Nevlyudov, I., Novoselov, S., Sukhachov, K. (2023), "Method of simultaneous localization and mapping for construction of 2.5D maps of the environment using ROS", *Innovative Technologies and Scientific Solutions for Industries*, No. 2 (24), P. 145–160. DOI: <https://doi.org/10.30837/ITSSI.2023.24.145>