

УДК 519.6

<http://dx.doi.org/10.30970/vam.2026.36.14059>

ІНТЕГРОВАНІЙ МЕТОД КОМП'ЮТЕРНОГО ЗОРУ ТА МАШИННОГО НАВЧАННЯ ДЛЯ МОДЕЛЮВАННЯ НАВІГАЦІЇ ЛІТАЛЬНИХ ПЛАТФОРМ З ОБМЕЖЕНИМИ ОБЧИСЛЮВАЛЬНИМИ РЕСУРСАМИ

С. Іванов, О.-Т. Кияк, А. Музичук

*Львівський національний університет імені Івана Франка,
вул. Університетська 1, Львів, 79007, Україна,
e-mail: serhii.ivanov@lnu.edu.ua, orest-teodor.kyiak@lnu.edu.ua
anatoliy.muzychuk@lnu.edu.ua*

Розглянуто поєднання візуальної одометрії з нейромережевим підходом для забезпечення автономної навігації за заданою супутниковою картою літальних платформ з обмеженими ресурсами бортового комп'ютера. Реалізація візуальної одометрії ґрунтується на послідовному зіставленні за допомогою методу SURF кадрів відеопотоку бортової камери, до яких попередньо застосовують гомографічне перетворення. Такий підхід дає змогу розглядати зображення сцени через “віртуальну” відеокамеру, оптична вісь якої спрямована вертикально вниз, причому необхідні значення кутів нахилу реальної відеокамери (піч і рол) обчислюють з рівнянь гомографії незалежно від параметрів польотного контролера. Перехід до зображень з віртуальної камери дає змогу простежити параметри їхньої зміни (переміщення, повороти, зміну масштабу) виключно із застосуванням афінних перетворень. Це в підсумку створює підстави для досягнення високої ефективності відповідного програмного забезпечення, яке обчислює траєкторію польоту платформи.

На практиці точність обчислення траєкторій залежить від структури сцени. Оскільки під час застосування візуальної одометрії обчислення координат має рекурентний характер, то відповідні похибки з часом нагромаджуються і можуть приводити до дрейфу локалізації. Тому в цій праці для уточнення координат платформи запропоновано використовувати нейронну мережу на основі архітектури Transformer. Побудована функція втрат такої моделі, за якою обчислюють міру подібності поточного відеокadra до блоків супутникової карти і формують так карту подібності. Розроблено алгоритм, який за картою подібності визначає кілька гіпотетичних траєкторій, які могли б привести платформу у найбільш імовірні точки сцени. Запропоновано інтегральний критерій для перевірки гіпотез і вибору оптимальної траєкторії. За координатами кінцевої точки обраної траєкторії уточнюють положення платформи на супутниковій карті місцевості. Отримана точка стає новим значенням вхідного параметра візуальної одометрії, з допомогою якої обчислюють новий фрагмент траєкторії польоту.

Наведено результати апробації запропонованого інтегрального підходу. Зокрема, на синтетичних даних, згенерованих з використанням бібліотеки OpenCV, продемонстровано високу точність і швидкодію вдосконаленої візуальної одометрії. Також на реальних даних, отриманих при обльоті заданої місцевості, отримано високу точність локалізації платформи на супутниковій карті в умовах обмежених обчислювальних ресурсів.

Ключові слова: візуальна одометрія, гомографічні перетворення, афінні перетворення, машинне навчання, обчислювальне моделювання, нейронні мережі, архітектура Transformer, прогнозні моделі, оцінка параметрів руху літальних апаратів, системи з обмеженими обчислювальними ресурсами.

1. ВСТУП

За умов обмежених обчислювальних ресурсів на борту БПЛА один з ефективних підходів до реалізації візуальної одометрії [1] ґрунтується на використанні афінного перетворення суміжних кадрів з відеопотоку бортової камери. Для його обчислення оптична вісь камери має бути перпендикулярною до площини земної поверхні. Такі зображення зазвичай називають ортогональними. У цьому випадку для літальної платформи кути вертикального нахилу (піч, pitch) і бокового (рол, roll) є нульовими. У реальних умовах складно забезпечити таку ідеальну фіксацію камери під час всієї дистанції польоту. У [2] було запропоновано “віртуальну” камеру, яка на підставі поточних значень зазначених кутів, отриманих з польотного контролера БПЛА, за допомогою гомографії перетворює зображення сцени на ортогональне з метою подальшого отримання характеристик руху з моделі афінного перетворення.

Розглянемо реалізацію візуальної одометрії шляхом отримання ортогонального зображення без додаткових даних з інших пристроїв. Це дасть змогу виконувати локалізацію БПЛА в режимі реального часу на бортовому комп’ютері. Беручи до уваги, що застосування візуальної одометрії характеризується нагромадженням похибки під час обчислення траєкторії БПЛА, інтегруємо цей підхід з нейромережовим, за допомогою якого будемо уточнювати траєкторії під час польоту.

2. МАТЕМАТИЧНА МОДЕЛЬ ВІРТУАЛЬНОЇ КАМЕРИ У ЗАДАЧІ ВІЗУАЛЬНОЇ ОДОМЕТРІЇ

Після зіставлення пар характерних точок на суміжних кадрах з відеопотоку відбувається двокроковий етап гомографії з метою корекції зображень з врахуванням кутів нахилу камери. Перший крок полягає в пошуку кутів пічу і ролу, на які потрібно повернути камеру, щоб отримати ортогональне зображення, другий – це безпосередній доворот камери. Тобто модель гомографії використовують двічі.

Нехай $P = \{p_i\}_{i=1}^n$ та $P' = \{p'_i\}_{i=1}^n$ – множини зіставлених точок на попередньому та поточному кадрах, відповідно, і $p_i := [u_i, v_i, 1]^T$ – позначення однорідних координат i -ї точки в системі координат зображення камери. Тоді математична модель проєктивного перетворення набуває вигляду [6]

$$\alpha_g p'_i = G p_i,$$

де $G \in \mathbb{R}^{3 \times 3}$ – матриця проєктивного перетворення; α_g – проєктивний множник для кожної точки, який визначає її глибину в сцені.

Матриця G задає відображення однієї площини на іншу в піксельній системі координат, заданій на сенсорі камери. Її можна виразити через налаштування камери (координати та орієнтацію) у глобальній системі координат та внутрішні параметри камери так:

$$G = \gamma K(R + t n^T) K^{-1}. \quad (1)$$

Тут t – вектор переміщення і R – матриця повороту, що описують зміну положення та орієнтації камери між суміжними кадрами; n – вектор нормалі до поверхні площини. Також тут K – матриця внутрішніх параметрів камери

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix},$$

де f_x, f_y – виражені в пікселях через фокусну відстань камери і розміри пікселів сенсора параметри, c_x, c_y – координати головної точки, яка є проєкцією оптичної осі на площину зображення, і γ – ненульовий коефіцієнт масштабу, з точністю до якого визначена гомографія.

Зауважимо, що відображення K переводить координати з тривимірного простору в евклідових координатах камери у двовимірну площину зображення. Тоді через неї можна визначити перетворення, яке задаватиме відображення в евклідовому просторі координат камери

$$\hat{H} = K^{-1}GK.$$

Тоді з врахуванням (1) матимемо відображення

$$\hat{H} = \gamma(R + t n^\top),$$

нормалізуючи яке, приходимо до перетворення

$$H := \frac{\hat{H}}{\gamma} = R + t n^\top.$$

Продемонструємо, що параметри руху камери можна отримати шляхом аналітичного розкладу матриці H [8, 10, 13], тобто виразити R, t, n безпосередньо як функції від елементів матриці H . Такий підхід є альтернативним до традиційного використання сингулярної декомпозиції матриць [4, 15].

Спочатку на основі H обчислюють симетричну допоміжну вироджену матрицю S

$$S = H^\top H - I = \begin{pmatrix} s_{11} & s_{12} & s_{13} \\ s_{12} & s_{22} & s_{23} \\ s_{13} & s_{23} & s_{33} \end{pmatrix}.$$

Тут $I \in \mathbb{R}^{3 \times 3}$ – одинична матриця. Для подальших розрахунків використовують значення $M_{S_{ij}}$ – двовимірних мінорів S з протилежним знаком.

Щоб знайти вектор нормалі n , було визначено три варіанти формул для уникнення проблеми ділення на нуль. Виведення виразів для обчислення n передбачає ділення на один із діагональних елементів S . Тому метод використовує варіант, де s_{ii} має найбільше абсолютне значення. Наприклад, нехай

$$|s_{11}| = \max_{1 \leq i \leq 3} |s_{ii}|, \quad s_{11} \neq 0,$$

тоді вектор нормалі n можна обчислити за формулою

$$n_e = \begin{bmatrix} s_{11} \\ s_{12} + k_e \sqrt{M_{S_{33}}} \\ s_{13} + k_e \epsilon_{23} \sqrt{M_{S_{22}}} \end{bmatrix}, \quad e \in \{a, b\}, \quad k_e = \begin{cases} 1, & e = a, \\ -1, & e = b. \end{cases}$$

де ϵ_{ij} – знак $M_{S_{ij}}$. Неоднозначність правильного варіанта серед n_a та n_b зумовлена недостатньою інформацією про геометрію сцени на цьому етапі алгоритму.

На підставі отриманих n_a та n_b обчислюють вектор переміщення t . Через неоднозначність існує кілька варіантів розв'язку. Отримавши можливі значення нормалі та вектора переміщення, обчислюють два можливі варіанти матриці повороту R

$$R_e = H \left(I - \frac{2}{\nu} t_e n_e^\top \right), \quad e = \{a, b\}.$$

Початкове розв'язання системи рівнянь для декомпозиції матриці гомографії H дає 8 можливих наборів параметрів $\{R, t, n\}$. З врахуванням базового фізичного обмеження – камера не може перебувати за площиною об'єкта – відкидаємо 4 розв'язки [8]. В результаті залишається 4 набори параметрів

$$\{R_a, t_a, n_a\}, \quad \{R_b, t_b, n_b\}, \quad \{R_a, -t_a, -n_a\}, \quad \{R_b, -t_b, -n_b\}$$

Усі характерні точки площини фізично мають бути розташовані перед камерою. Тому, щоб відсіяти частину неправильних розв'язків декомпозиції, в алгоритмі використано перевірку видимості важливих точок (visibility constraint) [11]. Такий підхід ґрунтується на аналізі знака проєкції координат характерних точок на нормаль площини з врахуванням матриці K [8]. Нехай $M = \{m_i\}_{i=1}^n$ та $M' = \{m'_i\}_{i=1}^n$ – множини проєкцій зіставлених ІЛР, визначені в системі координат камери

$$m_i = K^{-1}p_i, \quad m'_i = K^{-1}p'_i.$$

Вони виражають напрями променів, що з'єднують оптичний центр камери з відповідними точками тривимірної сцени.

Для кожного з 4 розв'язків декомпозиції гомографії $\{R, t, n\}$ виконуємо перевірку умови видимості усіх точок

$$m_i^\top n_k > 0, \\ m'_i{}^\top (R_k n_k) > 0.$$

У підсумку отримуємо набори параметрів $\{R, t, n\}$, для яких всі ІЛР сцени розташовані перед камерою в попередньому і поточному кадрах.

Однак навіть з врахуванням внутрішніх параметрів камери у деяких випадках неможливо однозначно визначити правильний результат декомпозиції гомографії. Зазвичай у таких ситуаціях залишаються два розв'язки, що відповідають тій самій гомографії, але є дзеркальними стосовно площини сцени.

Щоб усунути неоднозначність, треба використати додаткову інформацію про положення камери в сцені. Далі вважаємо, що площина поверхні сцени завжди має бути розташована перпендикулярно до оптичної осі камери. Тобто нормаль до площини має бути близькою до $n_{ideal} = [0, 0, 1]^\top$. На підставі цього припущення порівнюючи результати розкладу гомографії, вибираємо той, для якого евклідова відстань між обчисленою нормаллю та n_{ideal} є найменшою.

Після обчислення декомпозиції гомографії та визначення правильного варіанта розкладу $\{R, t, n\}$ відбувається пошук кутів нахилу камери стосовно площини. Кути пічу та ролу описують нахил оптичної осі камери стосовно площини сцени, що є основною причиною виникнення перспективних спотворень у зображенні. Компенсація цих кутів дає змогу вирівняти площину стосовно камери та усунути проєктивні деформації. Для обчислення пічу і ролу камери використовуємо знайдений вектор нормалі до площини.

Вектор нормалі $n = [n_x, n_y, n_z]^\top$ описує орієнтацію площини в системі координат камери на попередньому кадрі. Якщо оптична вісь камери перпендикулярна до поверхні, то нормаль має вигляд n_{ideal} . Будь-яке відхилення можна подати як послідовність поворотів навколо відповідних осей координат. Обчислення кутів відбувається з використанням функції $\arctan 2$. Піч отримуємо як кут між вектором нормалі та площиною XZ

$$\alpha = \arctan 2(n_y, \sqrt{n_x^2 + n_z^2}).$$

Рол визначається через відхилення нормалі вздовж осі X стосовно осі Z

$$\beta = \arctan 2(-n_x, n_z).$$

Однак кути нахилу камери можуть відрізнятися у різні моменти часу, зокрема між суміжними кадрами. Тому для поточного кадру нормаль трансформується за допомогою матриці повороту R , отриманої з декомпозиції гомографії

$$n' = Rn.$$

Це забезпечує узгодженість правильної орієнтації камери між кадрами. Так отримуємо кути нахилу камери на попередньому і поточному кадрах

$$\{\alpha, \beta\}, \quad \{\alpha', \beta'\} \quad (2)$$

Це дає змогу повернути відповідні зіставлені характерні точки, щоб отримати віртуальне ортогональне зображення площини.

На кожному кроці візуальної одометрії відбувається аналіз попереднього та поточного кадрів. Зіставлені точки інтересу P і P' характеризують площину сцени з врахуванням орієнтації камери (2) у відповідний момент часу. Щоб отримати віртуальне ортогональне зображення площини, для знайдених ІЛР на кожному кадрі застосовуємо матрицю гомографії, яка компенсує нахил камери у відповідний момент часу. Розгляньмо детальніше, як відбувається цей процес для попереднього кадру.

Перепроєктування точок на віртуальне ортогональне зображення площини виконуємо з використанням матриці гомографії H_{rect} . Вона виражає перетворення, яке потрібно застосувати до характерних точок P , щоб компенсувати нахил камери. Для обчислення H_{rect} спочатку знаходимо матрицю повороту $R_{rect} = R_x R_y$, яка виражає орієнтацію камери з врахуванням протилежних значень пічу і ролу, отриманих на основі декомпозиції [2]. Тоді матимемо

$$H_{rect} = K R_{rect} K^{-1}.$$

Оскільки P виражені в однорідних координатах, результатом застосування H_{rect} є точки інтересу в однорідних координатах

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ w \end{bmatrix} = H_{rect} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}.$$

Звідси отримуємо ІЛР на віртуальному ортогональному зображенні

$$\hat{u} = \frac{\tilde{u}}{w}, \quad \hat{v} = \frac{\tilde{v}}{w}.$$

Тобто матимемо множину точок $\hat{P} = \{\hat{p}_i\}_{i=1}^n$, $\hat{p}_i = [\hat{u}, \hat{v}, 1]^\top$ віртуальної камери, яка забезпечує інваріантність алгоритму до нахилів камери стосовно пічу та ролу. У підсумку це дає змогу застосовувати афінне перетворення для оцінки параметрів руху камери в сцені.

3. ОЦІНКА ГОМОГРАФІЇ НА СИНТЕТИЧНИХ ДАНИХ

Продемонструємо на синтетичних даних застосовність розглянутого алгоритму для обчислення траєкторій БпЛА. Для цього використаємо бібліотеку OpenCV і побудуємо модель камери [14] з відомими внутрішніми параметрами. Камеру розміщено на висоті 110 м над площиною спостереження, над якою відбувалась симуляція польоту. Оптична вісь камери була спрямована майже перпендикулярно з заданим відхиленням до поверхні площини. Відхилення моделювалось як коливання нахилу камери стосовно пічу на 3° - 4° протягом польоту. Зазначені параметри було обрано, оскільки вони є типовими для реальних умов польоту. Візуально такий нахил камери є непомітним, однак створює перспективне спотворення, яке потрібно враховувати. Симуляція польоту відбувалась вздовж прямої траєкторії довжиною 15 км зі швидкістю 30 км/год. Вважаємо, що на сцені задано перепади рельєфу до 20 м, як це зображено на рис. 1.

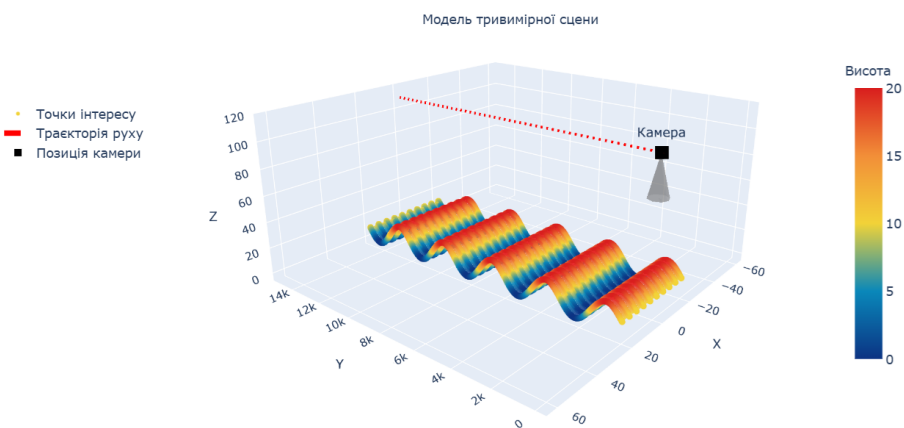


Рис. 1. Приклад візуалізації синтетичної тривимірної площини, над якою відбувається симуляція польоту змодельованої камери з використанням бібліотеки OpenCV

Для обчислення візуальної одометрії було згенеровано 4500 кадрів симуляції польоту над заданою сценою. Кожен кадр містив множину точок інтересу на площині, що потрапляли в поле зору камери у відповідний момент часу (рис. 2). Оскільки координати ІЛР були задані заздалегідь, то етап виявлення характерних точок методами SURF чи ORB не використовувався. Тому кожен крок одометрії складався зі зіставлення ІЛР у суміжних кадрах, компенсації перспективного спотворення й обчислення параметрів руху на основі афінного перетворення.

Результати обчислювального моделювання з використанням візуальної одометрії з корекцією за правилами гомографії демонструють високу точність. Зокрема, коливання нахилу камери стосовно пічу обчислено правильно (рис. 3,а), оскільки після довороту відбувається усунення перспективного спотворення. Також обчислені значення ролу дорівнює нулю, що відповідає польоту по прямолінійній траєкторії.

У візуальній одометрії масштаб виражає відношення відносного переміщення камери, яке алгоритм обчислює на основі зсуву точок між двома суміжними кад-

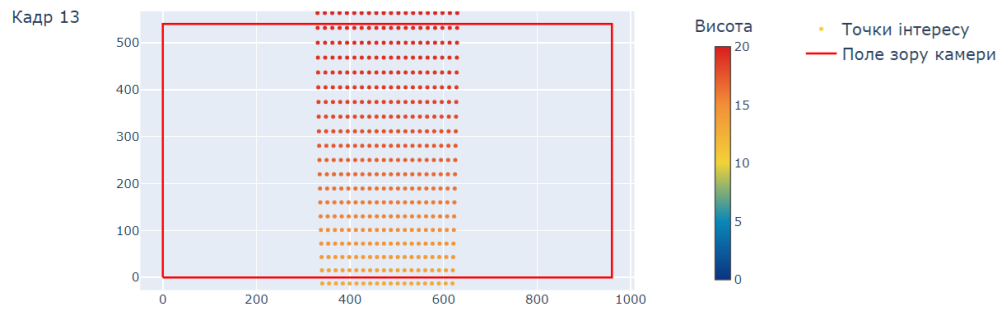


Рис. 2. Приклад візуалізації синтетичних точок інтересу у системі координат зображення під час симуляції польоту змодельованої камери над площиною

рами, до реального фізичного переміщення камери у сцені. На основі афінного перетворення обчислений масштаб (рис. 3б) відповідає зміні рельєфу. Основним критерієм оцінки алгоритму є точність обчисленої траєкторії. Її відхилення від заданої траєкторії з довжиною 15 км становить 87 м, тобто 0.58 %.

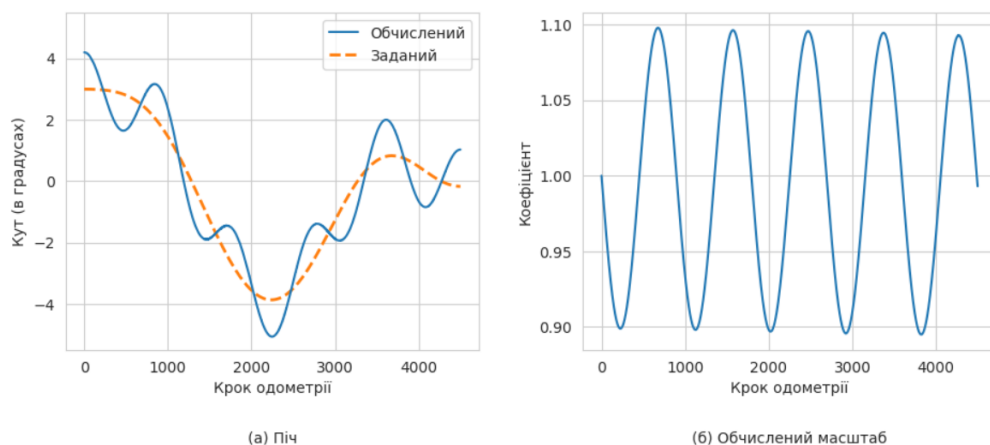


Рис. 3. Приклад результатів обчислення алгоритму з використанням корекції нахилу камери через гомографію

Попри високу точність результатів цього експерименту на ідеальних синтетичних даних, у реальних умовах з додатковими об'єктами на сцені, а також з використання дешевої відеокамери (з рядковим затвором, rolling shutter, яка дає нечітке розмите зображення) спостерігається нагромадження похибок під час обчислення траєкторій (див., наприклад, [1, 2]). Далі розглянемо застосування машинного навчання для уточнення розташування БпЛА на супутникових картах.

4. ЗАСТОСУВАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ЛОКАЛІЗАЦІЇ ЛІТАЛЬНОЇ ПЛАТФОРМИ

Нейромережевий підхід для визначення координат літальної платформи під час польоту ґрунтується на зіставленні зображення сцени з бортової камери з супутниковими зображеннями відповідної місцевості шляхом використання нейронної мережі з архітектурою Transformer. Її функція втрат спеціалізована на визначенні міри подібності зображень і фрагментів закодованої карти. На кожному кроці такого підходу результатом обчислень є ймовірна позиція БпЛА у світових координатах. Така властивість машинного навчання дає змогу уникати нагромадження похибки локалізації, яка притаманна візуальній одометрії.

Для побудови моделі скористаємося машинним навчанням на основі метрик (MBL, Metric Based Learning) [3], використовуючи нейронну мережу для проєкування $F : \mathbb{X} \rightarrow \mathbb{W}$ об'єктів деякого простору $\mathbb{X}$ у інший простір $\mathbb{W}$ (простір ознак), який допускає оцінку подібності двох своїх об'єктів стандартними методами.

У просторі ознак використовуватимемо позначення $f_x := F(x, \theta_F)$, $x \in \mathbb{X}$, де θ_F – параметри нейронної мережі F . Тоді для такої нейронної мережі цільова функція втрат

$$d = D(F(x_1, \theta_F), F(x_2, \theta_F), \theta_D), \quad x_1, x_2 \in \mathbb{X},$$

полягає у мінімізації результату d для подібних зображень x_1 і x_2 та максимізації для неподібних, відповідно. Тут θ_D – параметри моделі, що порівнює ознаки f_{x_1}, f_{x_2} .

Описану вище модель застосовуємо для випадку, коли вхідні зображення належать різним розподілам, а саме $\mathbb{X}_1$ – простір зображень з камери, $\mathbb{X}_2$ – простір зображень супутникової карти. Крім того, оскільки обидва об'єкти проєкуються на спільний простір ознак тією самою функцією, то вважаємо, що нейронна мережа під час тренування побудує таку функцію F , яка буде однаково опрацьовувати зображення з камери та супутникових карт: $F : \mathbb{X}_1 \cup \mathbb{X}_2 \rightarrow \mathbb{W}$.

На етапі роботи натренованої моделі обчислення подібності між векторами у просторі $\mathbb{W}$ виконуємо за формулою

$$S_C(f_x, f_y) = \frac{f_x \cdot f_y}{\|f_x\| \|f_y\|}, \quad x, y \in \mathbb{X}_1 \cup \mathbb{X}_2, \quad (3)$$

яку зазвичай називають “косинусною метрикою”.

Зауважимо, що визначення (3) не задовольняє означення метрики у метричних просторах, але є усталеним терміном у публікаціях і під час практичного застосування комп'ютерного зору.

Набір даних для тренування та тестування MBL-моделі складається з множини зображень з камери та зображення відповідної супутникової карти

$$\mathcal{D} = \{(s_i, l_i)\}_{i=1}^N,$$

де $s_i := (c_i, m_i)$, $i \in \overline{1, N}$, – пари зображень з камери літальної платформи c_i та відповідного фрагмента супутникової карти m_i ; l_i – мітка класу відповідного прикладу. Ця мітка класу пов'язана з просторовим положенням фрагмента супутникової карти та відповідним йому зображенням з камери.

Набір даних $\mathcal{D}$ створено шляхом запису реальних відеопотоків з бортової камери під час польоту літальної платформи (див. рис. 4), причому не передбачається щільний обліт усієї заданої локації.

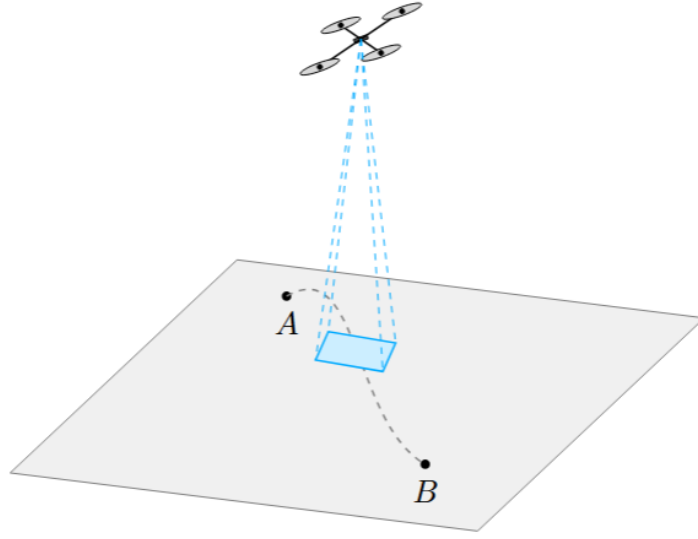


Рис. 4. Схема створення тренувального набору даних літальної платформи під час польоту над сценою з точки А в точку В; протягом всієї траєкторії були записані зображення з камери разом з координатами літальної платформи у відповідний момент часу

Зображення m_i , $i = \overline{1, N}$, супутникової карти було отримано шляхом кадрування неперервної карти так, щоб центр кадрованого фрагмента карти збігався з центром відповідного зображення. Розмір кадрування було підібрано, відповідно до FOV-параметрів камери та висоти польоту. Висота польоту становила 100 м з точністю до коливань, зумовлених турбулентністю повітря та вітру, та з врахуванням точності системи позиціонування (GPS) та датчиків висоти (барометр). Зображення карт було зібрано двома способами: з використанням провайдера супутникових карт Google Maps [5] та Here Maps [7], тобто отримано дві варіації однієї карти.

Для тренування моделі було зібрано набір даних з використанням 5 траєкторій польотів, що містить 9015 зображень з камери та 16736 відповідних їм фрагментів супутникових карт (різних дат і з різних супутників).

Нехай задано зображення з камери $c \in \mathbb{P}^{w \times h}$ і карту $M \in \mathbb{P}^{W \times H}$, $w < W$, $h < H$. Задача побудови карти подібності полягає в обчисленні такої матриці $P \in [0, 1]^{W \times H}$, за формулою

$$P_{i,j} = f(c, M'_{i,j}),$$

де f – модель, що обчислює подібність між двома зображеннями однакового розміру, $M'_{i,j}$ – фрагмент супутникової карти, який визначаємо так:

$$M'_{i,j} := M_{i - \lfloor \frac{w}{2} \rfloor : i + \lceil \frac{w}{2} \rceil, j - \lfloor \frac{h}{2} \rfloor : j + \lceil \frac{h}{2} \rceil}, \quad i = \overline{1, ?}, \quad j = \overline{1, ?}.$$

Тут вирази $\lfloor x \rfloor := \max\{n \in \mathbb{Z} \mid n \leq x\}$ та $\lceil x \rceil := \min\{n \in \mathbb{Z} \mid n \geq x\}$ означають правила заокруглення дійсних чисел до відповідних цілих.

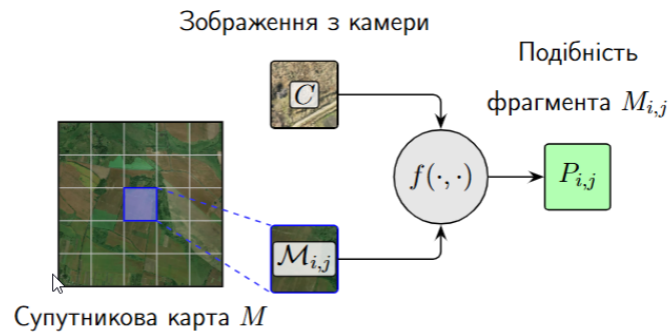


Рис. 5. Схема обчислення карти подібності P на основі супутникової карти M , зображення з камери c та функції обчислення подібності f

Отож $P_{i,j}$ репрезентує подібність зображення c до карти M в точці (i, j) за схемою, зображеною на рис. 5.

Зауважимо, що побудована карта подібності P не дає змогу точно визначити лише одну позицію на карті. Але за допомогою побудованої матриці можна окреслити області ймовірного перебування літальної платформи й оцінити ймовірність перебування літальної платформи у відповідних областях і далі висунути певні гіпотези про місце перебування літальної платформи. На рис. 6 показано приклад таких областей, яскравість кольору області на карті подібності характеризує ймовірність перебування БпЛА в цій області.



Рис. 6. Приклад побудованої карти подібності: а) зображення з камери; б) фрагмент супутникової карти; в) карта подібності

Відкидання та приймання гіпотез реалізовано шляхом аналізу зміни карти подібностей $P(t)$ у певні моменти часу t . Разом з умовою, що траєкторія польоту є неперервною, матимемо, що вона відповідатиме одній гіпотезі щодо позиції в конкретний момент часу. Ця обставина дає змогу відфільтрувати хибні гіпотези про локалізацію. Нехай кожна карта $P(t)$ пропонує множину гіпотез локалізації S_t . Тоді траєкторія руху належить графу, який складається з вершин S_t , з'єднаних в часі. Відповідно, можна відфільтрувати значну частину ребер, враховуючи апіорну інформацію про швидкість руху, а також відкинути усі траєкторії, які мають хоч одне відфільтроване ребро.

5. ІНТЕГРАЦІЯ ВІЗУАЛЬНОЇ ОДОМЕТРІЇ З МАШИННИМ НАВЧАННЯМ

Нехай функція $I(t) \in \mathbb{P}$, $t \geq 0$ задає відеопотік зображень сцени з бортової камери літальної платформи, а функція $p(t) \in \mathbb{R}^2$, $t \geq 0$ описує відповідну траєкторію літальної платформи з бігом часу. Під час обчислення траєкторій методом візуальної одометрії їхні координати лежать у системі координат зображення. У нейромережевому підході модель $g(I(t)) = P(t)$, $t \geq 0$ обчислює одразу ймовірність позицій на карті у геодезичних координатах. Модель $g : \mathbb{P}^{w \times h} \rightarrow \mathbb{R}^{W \times H}$ обчислює карту подібності P . Завдання побудови системи інтегрування двох зазначених підходів полягає в побудові системи позиціонування $F(f, g, I)(t)$, $t \geq 0$, яка об'єднує ці підходи в одній нейромережевій моделі.

Першим кроком до інтеграції двох методів є зведення їх у спільну систему координат. Оскільки нейромережевий підхід одразу прогнозує позицію у геодезичній системі координат, то для подальшої інтеграції потрібно результат роботи одометрії перевести з системи координат зображення у геодезичну систему координат. Отож нехай $\hat{p}(t) \in W$ є результатом візуальної одометрії в геодезичній системі координат.

Робота нейромережевого підходу разом з візуальною одометрією розпочинається в момент зникнення сигналу GPS. Першим кроком інтеграції є вибір калібраційної точки – позиції $\hat{p}(0)$ на карті у геодезичній системі координат, стосовно якої пізніше буде відбуватись уточнення поточних координат БПЛА за допомогою нейромережевого підходу. Стосовно цієї точки за алгоритмом візуальної одометрії обчислюють зміщення БПЛА, зберігаючи, крім координат поточної позиції, ще напрям руху літальної платформи. Його обчислюють за останніми даними GPS.

На підставі зображень сцени, які з деяким часовим кроком надходять з відеопотоку бортової камери, метод ВО в режимі реального часу обчислює зміщення точок у системі координат зображення, які за описаним вище алгоритмом проєктують у геодезичну систему координат, простежуючи поточну позицію $\hat{p}(t)$, $t \in (0, T]$ літальної платформи на карті відповідної місцевості.

Оскільки ВО нагромаджує з часом похибку, то реальна позиція літальної платформи перебуватиме у деякому околі точки $\hat{p}(t)$. Розглядатимемо область допустимих позицій платформи в околі поточної позиції $\hat{p}(t)$ у вигляді сектора кільця з внутрішнім $r_1(t) := (1 - k_1)|\hat{p}(t) - \hat{p}(0)|$ і зовнішнім $r_2(t) := (1 + k_2)|\hat{p}(t) - \hat{p}(0)|$ радіусами та з центром в калібраційній точці $\hat{p}(0)$. Значення гіперпараметрів $k_1 \in [0, 1]$, $k_2 \in [0, \infty]$ і кут сектора $\alpha \in [0, \pi]$ визначають експериментально, враховуючи характерну поведінку похибки ВО для типового середовища.

У межах цього сектора обчислимо метрики подібності $P(z, t)$ як подібність зображення $I(t)$ до фрагмента супутникової карти з центром в точці $z \in \mathbf{S}(t, \alpha)$. Отримана гіперповерхня $P(z, t)$ описує подібність місцевості на супутниковій карті до зображення з камери в заданий момент часу.

Нехай $\mathbf{E}$ – множина локальних максимумів поверхні $P(z, t)$ в секторі $\mathbf{S}(t, \alpha)$. Виберемо з неї підмножину $\mathbf{C}$, яка складається з K_p елементів, що відповідають позиціям з найбільшим значенням метрики подібності

$$\mathbf{C} = \{a_i \in \mathbf{E} \mid 1 \leq i \leq K_p\}.$$

Розглянемо критерій вибору з множини $\mathbf{C}$ однієї позиції, координати якої вва-

жатимемо уточненими значеннями координат реальної траєкторії в момент часу t .

З цією метою для кожної позиції $z \in \mathbf{C}$ розглянемо альтернативну траєкторію, що лежить на прямій між калібраційною точкою $\hat{p}(0)$ та $z \in \mathbf{C}$

$$\mathcal{T} := \{ \gamma_z(\tau) := (1 - \tau) \cdot \hat{p}(0) + \tau \cdot z \mid z \in \mathbf{C}, \tau \in [0, 1] \}.$$

Схему побудови таких траєкторій зображено на рис. 7. Зауважимо, що усі ці альтернативні траєкторії розпочинаються у калібраційній точці $\hat{p}(0)$, а закінчуються у точках $z \in \mathbf{C}$, які відрізняються між собою.

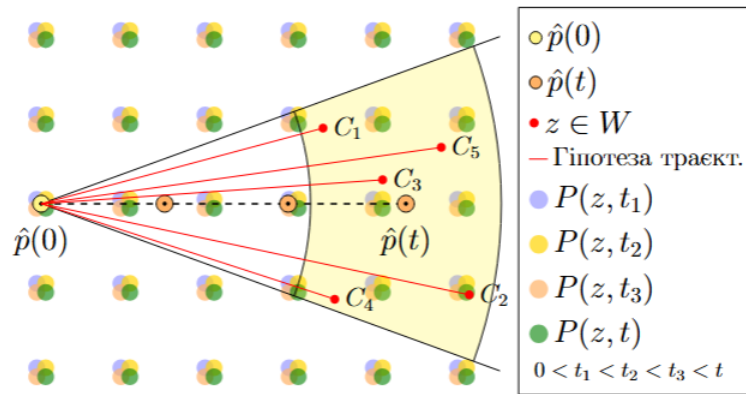


Рис. 7. Схема побудови гіпотез альтернативних траєкторій на основі неймережевого підходу

Для оцінки альтернативних траєкторій для кожної з них обчислимо вираз за формулою

$$E_z(t) := \frac{1}{t} \int_0^t P \left(\gamma_z \left(\frac{\tau}{t} \right), \tau \right) d\tau, \quad \gamma_z \in \mathcal{T}, \quad t \in [0, T],$$

яка враховує історію подібності усіх попередніх зображень сцени в часовому проміжку $[0, t]$ до відповідних фрагментів карт з центрами на кожній фіксованій траєкторії $\gamma_z(t)$. За точку уточнення виберемо таку $z^*(t)$, що

$$z^*(t) = \arg \max_{z \in \mathbf{C}} E_z(t), \quad t \in [0, T].$$

Після уточнення за допомогою повороту траєкторії спроекуємо поточну позицію $\hat{p}(t)$ у точку z^* , від якої далі продовжуємо обчислення траєкторії за алгоритмом візуальної одометрії.

Зауважимо, що описаний алгоритм уточнення траєкторії легко поширюється на випадки кусково-лінійних траєкторій.

Щоб продемонструвати обчислення траєкторій за допомогою описаного інтегрального методу, розглянемо прямолінійну (протяжністю ≈ 1 км) і криволінійну (протяжністю ≈ 2.2 км) траєкторії польотів над реальними сценами, Google-карти яких показано на рис. 8 і 9.

У першому випадку траєкторія проходить в основному над однорідним полем, перегинаючи автомобільну дорогу та смугу лісу. Зазначимо, що однорідність поля

зумовлює стосовно складний випадок для ВО через те, що немає орієнтирів. Попри це уточнення координат, було виконано згідно з описаним алгоритмом один раз після прольоту над смугою лісу. Після обчислення нової позиції БпЛА в цей момент алгоритм змінив поточну позицію БпЛА та його орієнтацію. Завдяки цьому, подальший рух на основі даних візуально збігся зі справжньою траєкторією польоту. Візуальне порівняння графіків демонструє також значну перевагу інтегрованого підходу над надзвичайною та над вдосконаленою одиметріями.

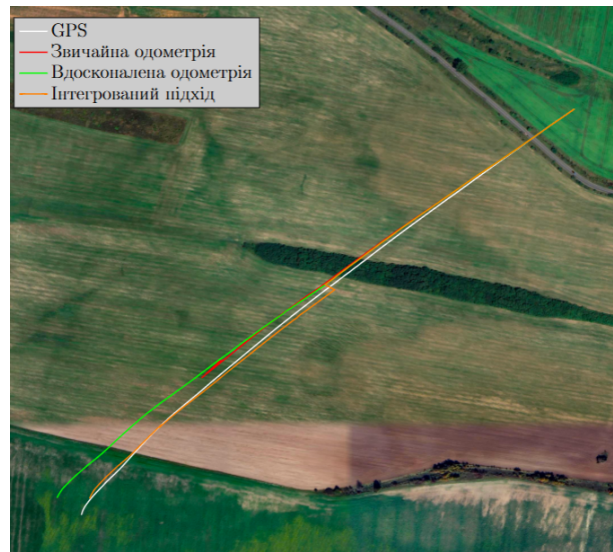


Рис. 8. Порівняння роботи інтегрованого методу з методом візуальної одиметрії

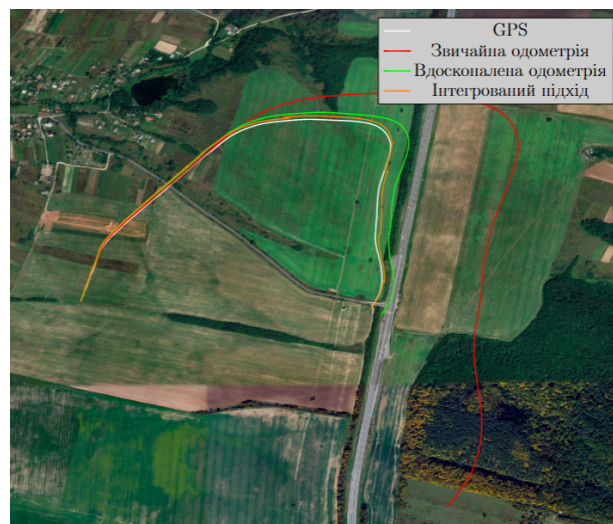


Рис. 9. Порівняння роботи інтегрованого методу з методом візуальної одиметрії

Складність другого прикладу (див. рис. 9) визначають кілька поворотів. Вони зумовлюють реініціалізацію калібраційної точки на кожному повороті, згідно з описаним алгоритмом. Графіки візуально демонструють, що інтегрований підхід успішно зменшив абсолютну локальну похибку обчисленої траєкторії. На завершальному етапі польоту відхилення від заданої траєкторії протяжністю ≈ 2.2 км становило 5.39 м, тоді як похибка вдосконаленої одометрії дорівнювала 49.13 м. Зазначимо, що ця величина для звичайної одометрії становила 894.83 м.

6. ВИСНОВКИ

Розглянута корекція зображень сцени, яка ґрунтується на аналітичному підході щодо знаходження кутів нахилу камери і переході до віртуальних ортогональних зображень у візуальній одометрії, дає змогу компенсувати перспективне спотворення без використання додаткових бортових сенсорів. Інтегрування такого підходу з машинним навчанням дає змогу будувати високопродуктивне програмне забезпечення, придатне для виконання задачі локалізації БпЛА на супутникових картах в реальному часі за умови обмежених обчислювальних ресурсів бортового комп'ютера. Результати апробації такого програмного забезпечення демонструють високу точність обчислення траєкторій польоту БпЛА у разі моделювання на синтетичних, і на реальних даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Іванов С.А. Про використання кластерної вибірки в алгоритмах візуальної одометрії / С.А. Іванов, А.О. Музичук // Вісник ЛНУ. Серія прикл. матем. та інформ. – 2023. – Вип. 31. – DOI: <http://dx.doi.org/10.30970/vam.2023.31.11972>.
2. Іванов С.А. Про застосування методів візуальної одометрії для автономної навігації літальних апаратів / С.А. Іванов, А.О. Музичук // Вісник ЛНУ. Серія прикл. матем. та інформ. – 2024. – Вип. 32. – DOI: <http://dx.doi.org/10.30970/vam.2024.32.12418>.
3. Bromley J. Signature Verification using a “Siamese” Time Delay Neural Network / J. Bromley, J. Bentz, L. Bottou, I. Guyon, Y. Lecun, C. Moore, E. Sackinger, R. Shah // International Journal of Pattern Recognition and Artificial Intelligence. – 1993. – Vol. 7. – P. 25. – DOI: <https://doi.org/10.1142/S0218001493000339>.
4. Faugeras O. Motion and structure from motion in a piecewise planar environment / O. Faugeras, F. Lustman // International Journal of Pattern Recognition and Artificial Intelligence. – 1988. – Vol. 2 (3). – P. 485–508.
5. Google Maps: Картографічний вебсервіс. – URL: <https://www.google.com/maps>.
6. Hartley R. Multiple View Geometry in Computer Vision (2nd ed.). / R. Hartley, A. Zisserman. – Cambridge University Press, 2004.
7. Here Maps Картографічний вебсервіс. – URL: <https://maps.here.com>
8. Malis E. Deeper understanding of the homography decomposition for vision-based control / E. Malis, M. Vargas // INRIA Research Report. – 2007. – Vol. RR-6303. – P. 90.
9. OpenCV: Camera Calibration and 3D Reconstruction. – URL: https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html.
10. OpenCV: decomposeHomographyMat. – URL: https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html#ga7f60bdf78833d1e3fd6d9d0fd538d92.
11. OpenCV: filterHomographyDecompByVisibleRefpnts. – URL: https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html#ga32f867159200f7bd55e72dca92d8494c.
12. OpenCV: findHomography. – URL: https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html#ga4abc2ece9fab9398f2e560d53c8c9780.
13. OpenCV Homography Decomposition Implementation (GitHub). – URL: https://github.com/opencv/opencv/blob/4.x/modules/calib3d/src/homography_decomp.cpp.

14. OpenCV: projectPoints. – URL: https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html#ga1019495a2c8d1743ed5cc23fa0daff8c.
15. Zhang Z. 3D Reconstruction based on homography mapping / Z. Zhang, A.R. Hanson // Proc. ARPA96. – 1996. – P. 1007–1012.
16. Zhang Z. A flexible new technique for camera calibration / Z. Zhang // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2000. – Vol. 22 (11). – P. 1330–1334.

Стаття: надійшла до редколегії 14.01.2026

доопрацьована 25.02.2026

прийнята до друку 04.03.2026

AN INTEGRATED COMPUTER VISION AND MACHINE LEARNING METHOD FOR MODELLING THE NAVIGATION OF AERIAL VEHICLES WITH LIMITED COMPUTATIONAL RESOURCES

S. Ivanov, O. Kyiak, A. Muzychuk

Ivan Franko National University of Lviv,

1, Universytetska str., 79007, Lviv, Ukraine,

*e-mail: serhii.ivanov@lnu.edu.ua, orest-teodor.kyiak@lnu.edu.ua
anatoliy.muzychuk@lnu.edu.ua*

This paper examines the combination of visual odometry with a neural network approach to enable autonomous navigation of aerial platforms with limited on-board computer resources using a predefined satellite map. The implementation of visual odometry is based on sequential matching, using the SURF method, of frames from the on-board camera's video stream, to which a homographic transformation is first applied. This approach allows the scene to be viewed through a “virtual” video camera, the optical axis of which is directed vertically downwards, with the required pitch and roll angles of the real video camera being calculated from the homography equations independently of the flight controller parameters. The transition to images from a virtual camera allows the parameters of their transformation (translation, rotation, scaling) to be tracked exclusively using affine transformations. This ultimately provides the basis for achieving high efficiency in the corresponding software that calculates the platform's flight trajectory.

In practice, the accuracy of trajectory calculation depends on the scene structure. Since, when using visual odometry, the calculation of coordinates is of a recursive nature, the corresponding errors accumulate over time and can lead to localisation drift. Therefore, in this work, we propose using a neural network based on the Transformer architecture to refine the platform's coordinates. A loss function has been constructed for such a model, which calculates the similarity of the current video frame to blocks of the satellite map and thus forms a similarity map. An algorithm has been developed which, based on the similarity map, determines several hypothetical trajectories that could lead the platform to the most probable points in the scene. An integrated criterion is proposed for testing hypotheses and selecting the optimal trajectory. The platform's position on the satellite map of the area is refined using the coordinates of the end point of the selected trajectory. The resulting point becomes a new value for the input parameter of visual odometry, which is used to calculate a new segment of the flight trajectory. The paper presents the results of testing the proposed integrated approach. In particular, on synthetic data generated using the OpenCV library, the high accuracy and speed of the improved odometry are demonstrated. Furthermore, using real data obtained during a flight over a specified area, high accuracy in localising the platform on a satellite map was achieved under conditions of limited computational resources.

Key words: visual odometry, homographic transformation, affine transformations, machine learning, computational modeling, neural networks, transformer architecture, predictive models, aerial vehicle motion estimation, resource-constrained systems.