

ФІЛЬ Дмитро Тарасович

**Метод та програмне забезпечення визначення
віддалі до перешкод мобільного робота на
основі обробки стерео-зображень / Method and
Software for Determining the Distance to Obstacles
of a Mobile Robot based on Stereo Image
Processing**

спеціальність: 121 - Інженерія програмного забезпечення
освітньо-професійна програма - Інженерія програмного забезпечення

Кваліфікаційна робота

Виконав студент групи ІПЗм-21
Д. Т. Філь

Науковий керівник:
д. філос. О. А. Папа

Кваліфікаційну роботу
допущено до захисту:

" ____ " _____ 20 ____ р.

Завідувач кафедри
_____ **А. В. Пукас**

ВСТУП

Актуальність. Розвиток сучасних роботів стрімко починає охоплювати принципово нові сфери застосування, пов'язані з виконанням робіт у невизначених і важких умовах - від космічних і глибоководних досліджень, до обслуговування атомних електростанцій, ліквідації наслідків аварій і техногенних катастрофи, боротьба з тероризмом і використання в інтересах збройних сил, поліції та інших спецпідрозділів, виконання надскладних медичних операцій, автоматизація сфери обслуговування, організація життя та розваг.

В останні роки цифрова обробка та цифровий аналіз зображень знайти безліч застосувань у різних галузях науки і техніки, таких як такі як інтелектуальні роботизовані конструкції, промислові системи управління, системи керування мобільними пристроями, обробка даних дистанційне зондування, біомедичні дослідження, нов технологія обробки документів та багато іншого .

Технологія технічного зору широко використовується в робототехніці. Одне з найскладніших завдань з технічної точки зору його застосування до роботів – розпізнавання, яке включає виявлення та ідентифікація об'єктів, побудова просторових карт.

Складність проблеми «розуміння зображення» пояснюється тим, що її алгоритмічна складова величезна. складніше будь-якого традиційного завдання , алгоритм якого простий піддається алгоритмізації. Комп'ютер, який нарівні грає в шахи світові герої, вже створені, і комп'ютерна програма, яка « розуміє». сцени ще не видно.

У цій роботі розглядаються різноманітні складності, які вже існують рішення, які включають навігаційні системи технічного зору мобільні роботи. Включає розглянуті питання, пов'язані з розвитком системи технічного зору для автономного керування робототехнічними системами .

У цій статті запропоновано алгоритм обчислення відстані до спостережуваного об'єкта на основі зображень із стереопари. Актуальність даної роботи полягає в тому визначення відстані до об'єкта за фотографіями з двох камер - одна з ключових функцій систем комп'ютерного зору . Це робиться в лабораторіях по всьому світу дослідження практичного використання стереозору для отримання інформації про навколишній світ. Одним із застосувань стереозору є пересування мобільних роботів.

Узгодження роботи з науковими програмами, проектами, темами

Проведений напрямок дослідження безпосередньо відповідає науковому напрямку кафедри «Комп'ютерні науки» Тернопільського національного економічного університету.

Мета і завдання дослідження

Метою даної роботи є розробка алгоритму визначення відстані до перешкод мобільного робота на основі обробки стереозображення.

Основними видами наукової діяльності є:

- аналіз проблем використання комп'ютерного зору, визначення основних напрямів їх удосконалення;
- проаналізувати з точки зору керування рухом мобільного робота та провести класифікацію потенційних перешкод , що виникають у процесі руху мобільного робота .
- дослідження методів і алгоритмів визначення параметрів об'єкта та відстані до нього на основі обробки зображень;
- аналіз методів визначення відстані до перешкод під час руху мобільного робота;
- розробка алгоритму розрахунку відстані до спостережуваного об'єкта за зображеннями зі стереопари ;
- побудова схем у стереомоделі реалізації обраного алгоритму;
- процедури калібрування, спотворення та обробки зображень;
- побудова карти глибини та побудова математичної моделі формування стереопотоку ;

- виконати програмну реалізацію системи визначення відстані до перешкод мобільного робота ;
- проведення експериментальних досліджень і розробка рекомендацій щодо практичного застосування розроблених методів і алгоритмів.

Об'єкт і предмет дослідження. Метою дослідження є процес визначення відстані до перешкод мобільного робота на основі обробки стереозображення . У центрі уваги цього дослідження методи та програмно-технічні методи визначення відстані до перешкод мобільного робота .

Методи дослідження. Теоретичні дослідження розробки моделі користувача базуються на застосуванні системного аналізу, підходів до функціонального моделювання, когнітивної інженерії, штучного інтелекту, теорії множин, теорії графів, теорії нечіткого логічного висновку, теорії алгоритму та об'єктно-орієнтованого проектування.

Наукова ефективність отриманих результатів. Пропонується метод визначати відстань до спостережуваного об'єкта за зображеннями зі стереопари, що, на відміну від відомих рішень, дозволяє знизити вимоги до пропускної здатності каналу передачі даних.

практичне застосування отриманих результатів полягає в розробці програмного забезпечення , яке можна інтегрувати в інтелектуальні системи та управління автономними мобільними роботами

Індивідуальний внесок магістранта . Усі результати отримані автором самостійно.

Отримання результатів. Основні критерії магістерської роботи були розглянуті у Всеукраїнській школі-семінарі для молодих вчених та студентів «Комп'ютерні інформаційні технології», яка відбулася 29 листопада 2019 року в м. Тернополі за матеріалами ТНУ.

Публікації . Кравчук Д.В., а засоби. Математичне та програмне забезпечення семантичного захисту інформації в корпоративних системах // Всеукраїнський шкільний матеріал-семінар молодих учених та студентів

«Комп'ютерні інформаційні технології». — Циркуляр: ФОП Шпак В.Б., 2019
. - С.13.

РОЗДІЛ ПОСТАНОВКА ПРОБЛЕМИ ДОСЛІДЖЕННЯ

1.1. Функції та технологія комп'ютерного зору

Комп'ютерний зір є одним з найперспективніших способів автоматизації будь-яких дій з використанням комп'ютерної техніки та робототехніки. У найзагальнішому вигляді системи технічного зору мають на увазі перетворення даних, отриманих від пристроїв захоплення зображення, і подальші оперативні операції на основі цих даних.

Проблема технічного зору дуже приваблива для сучасних дослідників з тієї причини, що апаратні можливості, які використовуються в цій області, досягли точки, коли вони вже багато в чому наближаються до людських «технічних атрибутів».

Розширення більшості сенсорів для прийому відеоданих істотно пропорційно кількості елементів сітківки людського ока, а можливості комп'ютерів і спеціалізованих процесорів наближаються до характеристик «обчислювальної потужності», використовуваної для обробки зображень в мозок.

Зі свого боку, біомеханіка найбільше підходить до розробки складних механічних пристроїв, достатніх для імітації рухової діяльності людини для управління різними технічними системами. І отже, на шляху до реалізації заповітної естетичної мрії майбутнього – створення складних автономних роботизованих конструкцій, «розумних машин», що працюють у масштабі реального часу, – виникає потреба розв'язати саме одне завдання «тільки одне». основний - . розробка методів і алгоритмів «розуміння» зображень. Однак це завдання в багатьох випадках виявляється дуже складним [1].

Основною проблемою є відсутність будь-якої «каузальної» або динамічної моделі формування семантичного змісту інформації про зображення, в тому сенсі, що цей семантичний зміст інформації не виникає під впливом будь-яких законів фізичного, що описується математичними рівняннями. Інформація про зображення виражається у формі нескінченної різноманітності геометричних форм, для яких моделі створення можуть просто не існувати.

Найважче завдання – «розібратися» з об'єктами в обсерваторії. Виявлення та ідентифікація багатьох типів таких об'єктів, наприклад, будівель і доріг на аерофотознімках, навіть стали окремими напрямками дослідження. Так, за останні роки кілька великих міжнародних конференцій були присвячені виключно проблемі фотодруку будівель, привносячи лише кілька нових проблем на додаток до існуючих [2].

Таким чином, слід зазначити, що загальна теорія «образного сприйняття» в останні 30-40 років ще не вийшла з молодості, і її фіксований стан сьогодні є синтезом ряду теоретичних робіт, нерозв'язаних одна з одною. з одного боку, і велика кількість припущень і підходів, далеких від остаточної форми добре розробленої теорії, з іншого. Теорія технічного зору надає ряд різних модельних описів спостережуваних об'єктів, які можна використовувати для їх виявлення та вимірювання. У літературі описано широкий спектр таких моделей - від простих визначень ознак до вузькоспеціалізованих і складних структурних моделей.

В останні десять-п'ятнадцять років в алгоритмічному аспекті прийнято розглядати послідовності дій обробки зображень у поєднанні з так званою модульною парадигмою [5]. Ця парадигма, запропонована Д. Марром на основі лонгітюдного дослідження зорових механізмів людини, має на увазі, що обробка зображення повинна базуватися на кількох послідовних рівнях цієї інформаційної черги по висхідній: від «зображувального» представлення об'єктів (растрове зображення, неструктуроване зображення). інформації) до

їх символічного представлення (векторні та атрибутивні дані в структурованій формі, реляційні структури тощо).

Відповідно, в області машинного зору прийнято виділяти наступні основні етапи обробки даних:

- 1) попередня обробка зображення;
- 2) категоризація;
- 3) завдання геометричної форми;
- 4) визначити структуру і семантику рівняння [5].

Рівні обслуговування, пов'язані з цими етапами, зазвичай називають у такому порядку: обробка низького рівня, обробка середнього рівня, обробка високого рівня.

У той час як алгоритми обробки низького рівня (проста фільтрація шуму, обробка гістограм) можна вважати добре розробленими та широко вивченими, алгоритми середнього рівня (сегментація) продовжують бути в центрі інженерних і дослідницьких зусиль сьогодні. . . . За останні роки було досягнуто значного прогресу щодо відображення проблем точок і поверхонь зображень, виділення функцій у невеликих 3D-макетах точок з високою точністю, передбачаючи належне моделювання та калібрування датчиків та їх комбінацій, створення простих геометричних форм типу «точка», «край», «пляма», «пряма лінія», «кут» [7].

Передові методи обробки, пов'язані з «розумінням зображення», все ще знаходяться на ранній стадії розробки та представляють «виклик» для дослідницького співтовариства в галузі технічного (комп'ютерного) зору та інтелекту. Звичайно, перспектива створення майбутніх поколінь «розумних машин» багато в чому залежить від продовження розвитку цього циклу алгоритмів.

Технічне бачення відноситься до прикладної дисципліни інженерії. Кожне прикладне технічне дослідження відрізняється від основного теоретичного дослідження наявністю певних практичних обмежень і характеристик. Розглянемо вимоги до методів і алгоритмів машинного зору на

прикладі дуже специфічного класу алгоритмів – алгоритмів виявлення об'єктів на зображеннях. Є три основних типи вимог:

- 1) міцність;
- 2) точність;
- 3) здійснення розрахунку.

Надійність Усі розроблені алгоритми, що використовуються для вирішення великих прикладних задач технічного зору, повинні бути застосовні до реалістичного моделювання. Фактори, що впливають на процес обробки та аналізу зображень, дуже різноманітні та їх важко формалізувати.

Основні фактори впливу:

- 1) перешкоди та «шуми» – мають багато типів джерел виникнення, до яких відносяться сенсорні недосконалості апаратури та цифрового фотообладнання, складні умови зйомки, недостатнє освітлення та погане читання;
- 2) складний фон, на якому має відбуватися виявлення об'єкта;
- 3) ефект гальмування одних факторів іншими факторами, як правило, непередбачуваного характеру;
- 4) спотворюють оптичні ефекти у вигляді різноманітних розфокусувань, лінзових спотворень, перспективних спотворень тощо;
- 5) наслідки різких змін світла, відблиски, тіні;
- 6) варіації та зміни в самих об'єктах – динамічна структура (як у рядках тексту, мобільних номерах або штрих-кодах), можливі помилки, тимчасові зміни форми (розширення елементів). , рух механічних частин машин);
- 7) ефекти модифікації навколишнього середовища між датчиками та об'єктами спостереження – дим, дощ, пил, штучні перешкоди та багато інших [1,2];

Специфіка. Другу за важливістю властивість, якою, як правило, повинні володіти алгоритми об'єктів на зображеннях, можна визначити як точну локалізацію об'єктів, що підлягають виявленню, або контури об'єктів , які підлягають вимірюванню. Це означає, що необхідно не тільки виділити об'єкт,

але і чітко вказати його положення і обмеження будь-яким способом в системі координат діаграми (або виду).

Характерні символи - штрих-код на зім'ятій упаковці. У цих випадках локалізація не має значення. У найпростішому випадку, якщо параметри геометрії викривлення вказуються аналітично явно, точна локалізація може розумітися як знання про положення будь-якого характерного розташування об'єкта та параметри геометрії для скручування (крени, кривої тощо).

Локальні несправності прийнято поділяти на дві групи - нормальні та аномальні.

Стандартна похибка - це правильне розміщення об'єкта з деякою позиційною або параметричною неточністю, що характеризується кількісними оцінками.

Рідкісні помилки включають ситуацію плутанини або появу артефактів, пов'язаних із фатальними кількісними помилками позиціонування або просто неправильним виявленням. Вимоги щодо видалення або оцінки ступеня рідкісних помилок складають дуже важливу частину вимог до алгоритмів розпізнавання.

Ми також зазначаємо, що вимоги до точності також застосовуються до алгоритмів низькорівневої обробки зображень. Наприклад, у більшості вимірювальних завдань фільтри, що усувають перешкоди, не повинні істотно змінювати видиме розташування контурів об'єкта. Вимога «кінцевого збереження» є суворою вимогою «точності» для алгоритмів низького рівня. Це пов'язано з тим, що якщо дані «пошкоджуються» на нижчому рівні обробки, то на вищих рівнях аналізу навіть самі статистичні методи більше не будуть корисними, а це означає, що робота не буде вирішена тим, що необхідно для роботи.

Застосування бухгалтерського обліку. Комп'ютерні технології в останні десятиліття двадцятого століття та перші роки двадцять першого століття пройшли довгий шлях розвитку, була розроблена велика спеціалізована процесорна база для обробки зображень, особливо різних програм, особливо

промислових програм реального часу, але незважаючи на це, характеристики комп'ютерів і їх загальні характеристики ще далекі від попиту.

Важлива відмінність підходів до обробки та аналізу зображень у порівнянні з уже сегментованими завданнями розпізнавання або інтерпретації зображень полягає в тому, що виявлення та вимірювання в практичних завданнях завжди пов'язані з процесом пошуку об'єкта. . . . Це застосування механізму пошуку об'єктів, пов'язаного із загрозою лавиноподібного сплеску потрібної кількості обчислень. Наприклад, якщо ви шукаєте об'єкт на основі порівняння з растровим посиланням або зображенням. Навіть враховуючи експоненційне збільшення потужності сучасних комп'ютерів, такі обсяги обчислень займають секунди машинного часу, що значно перевищує потреби систем розпізнавання в реальному часі, призначених для таких завдань, як ходьба, подорожі та безпосереднє керування тестуванням, обробка машинозчитуваного документи тощо d.

Таким чином, обчислювальна реалізація алгоритмів машинного зору все ще є одним із найважливіших міркувань під час їх розробки.

1.2. Завдання керування мобільністю мобільних роботів та класифікація потенційних перешкод

Розглянемо задачу керування рухом мобільного робота, яка може бути формалізована наступним чином. Існує кусково-лінійна траєкторія від початкової точки до цілі. Її можна обробляти в режимі реального часу за допомогою алгоритму A* (A-star). Навігаційна система мобільного робота відстежує його положення і обчислює координати поточної точки відліку, в якій робот повинен рухатися, уникаючи зіткнень з перешкодами (фіксованими і динамічними).

Стационарні шлагбауми (рисунок 1.1). Розглядайте розв'язок задачі в умовах фіксованого простору як простий. В його основі лежить метод практичного методу [2]. У цьому випадку активний шлях – це відрізок лінії, що з'єднує поточне положення робота з поточною опорною точкою.

Другий залишок кусково-лінійної траєкторії (рисунок 1.2) використовувався як точка відліку, оскільки однієї зміни напрямку траєкторії з різним кроком 45° відносно положення робота недостатньо для довірчого визначення тенденції для певної зміни напрямку. Друга зміна напрямку свідчить про стійкість зміни заданого напрямку руху. В іншому випадку робот продовжує рух у вибраному напрямку.

Відстань R між орієнтиром на траєкторії та поточним положенням робота визначається його динамікою. Зі сказаного вище, значення R обмежується максимальною швидкістю ходьби $v_c + v_a \Delta t$, яку може досягти робот протягом інтервалу часу Δt .

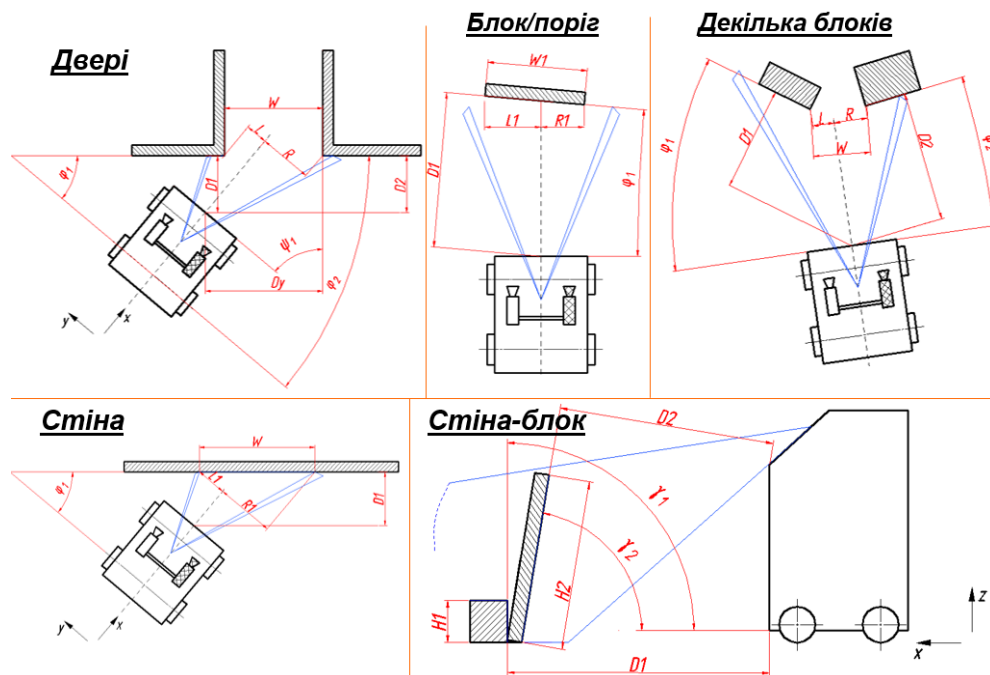


Рис. 1. 1 . Визначення загального набору фіксованих обмежень

$$R_{max}(V_c) = (V_c + v_a \Delta t) T_{max}, \quad (1.1)$$

де

v_a - максимально допустиме прискорення робота; V_c - поточна швидкість руху робота; T_{max} - час досягнення роботом найближчої перешкоди, що знаходиться на S_l відстані на високій швидкості

$$v_{max} \cdot T_{max} = S_l / v_{max}, \quad (1.2)$$

тобто ефективна довжина шляху залежить від швидкості робота.

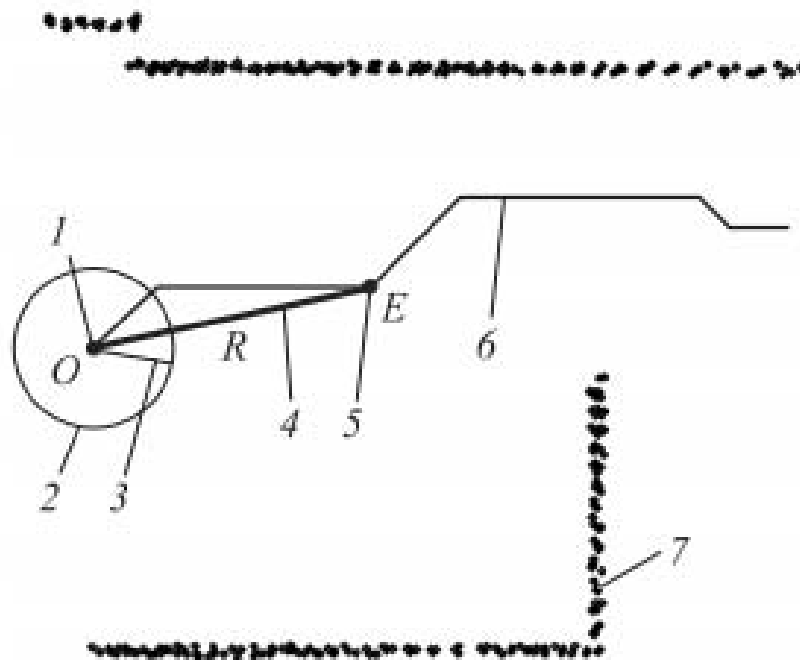


Рис. 1.2. Виявлення контрольної точки: 1 – місцезнаходження робота, 2 – мобільний робот, 3 – поточний шлях робота, 4 – активний шлях, 5 – контрольна точка, заданий шлях, 7 – перешкоди

Якщо зміна орієнтації виявлена в безпосередній близькості від робота (наприклад, клітинка карти поблизу поточного місцезнаходження робота), ця зміна не має значення для робота. Таким чином, відстань між вашою точкою відліку на траєкторії та поточним положенням робота вимірюється від землі в постійний час, R_{min} що відфільтровує невеликі зміни напрямку його течії. Такі

зміни можуть затримувати робота, але вони зовсім не змінюють його положення. Що дуже вдало встановить значення R_{min} відповідно до максимального часу вибуху:

$$R_{min} = \left(V_{max} T_{bmax} - \frac{1}{2} v_b \Delta t, \right) T^2_{bmax}, \quad (1.3)$$

З цього впливає низька швидкість руху

$$v_{min} = R_{min}/T_{max}, \quad (1.4)$$

Потім шлях активного шляху дає координати опорної точки в тому напрямку, куди повинен цілитися робот, рухаючись у заданому напрямку. При цьому траєкторія плавна і робот може зіткнутися з перешкодою. Також можливі зіткнення, якщо в алгоритмі пошуку шляху неправильно вказано параметри робота. Щоб уникнути подібних ситуацій, дієвий напрямок методу було модифіковано. Суть зміни полягає в тому, щоб штучно перемістити дослідницьку територію із «забороненої», коли вона туди потрапляє. Зверніть увагу, як обчислити виправлену площу опорної точки.

Нехай буде схема кімнати (рисунок 1.3). На ньому зображено траєкторію робота 3 і точку відліку E , координати якої розраховані методом активної траєкторії. Біля мобільного робота є перешкоди, позначені на карті осередками, що займають 5. Межа робота визначена колом, її радіус W визначається розмірами робота, зоною комфорту та розміром карти заповнення клітини.

Спочатку вказується, наприклад, область, де слід враховувати бар'єри $N_c \times N_c$ клітинки. Потім кожна клітинка в межах цієї області перевіряється на наявність перешкод. Після того, як перешкоди є визначені, координати точки відліку обчислюються шляхом розгляду такі умови:

- робот не повинен стикатися з перешкодою в новому дослідницькому середовищі;

- необхідно підтримувати ефективну довжину шляху.

Розглянемо трикутник OEP (рис. 1.3, а). Тут точка O — центр робота; E - дослідницька ділянка; P - центр комірки з бар'єром; OP — ефективний шлях довжини R . Спочатку ми оцінюємо ймовірність того, що робот зіткнеться з перешкодою в цільовій зоні. Для цього розраховуємо відстань d між точкою відліку E і точкою P :

$$d = |EP| = |OE - OP|, \quad (1,5)$$

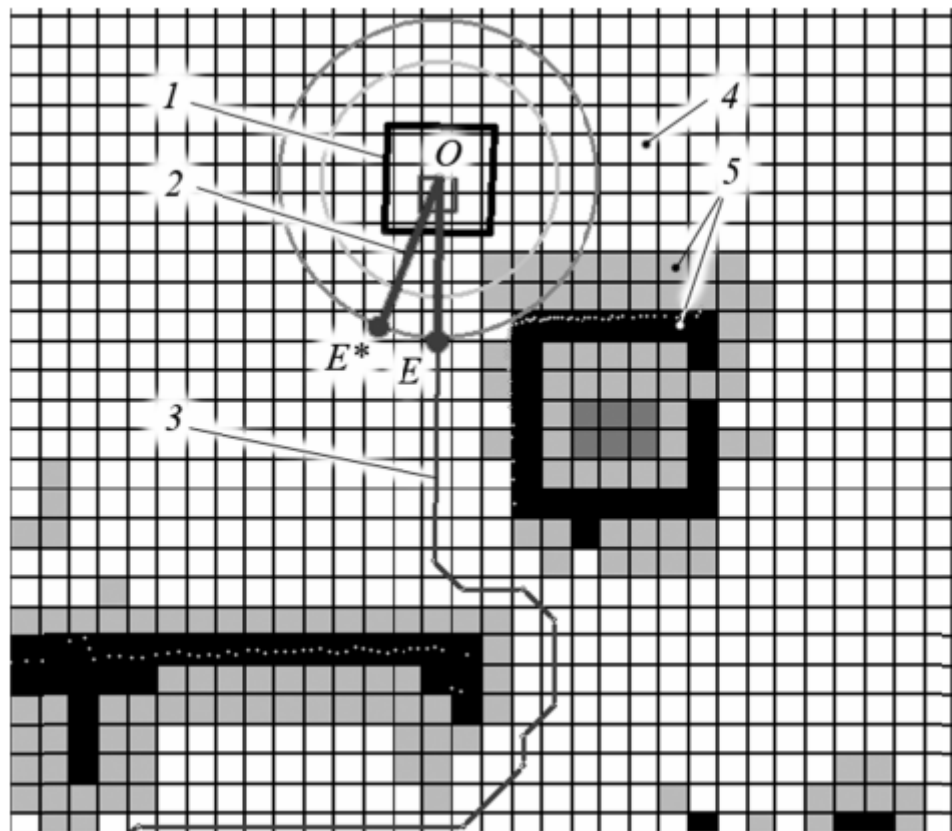


Рис. 1.3. Результати визначення місця дослідження: 1 – мобільний робот; 2 – ефективний метод; 3 – задана траєкторія; 4 – вільні комірки; 5 – резидентні комірки

Коли $d \geq W$, перешкода знаходиться далеко від робота та ігнорується. В іншому випадку вибирається нове положення контрольної точки. Розглянемо екстремальний випадок, коли вісь, визначена роботом, торкається $d = W$.перешкод. З рисунка 3 видно, що нова контрольна точка E^* повинна знаходитися на перетині осей (P, W) і (O, R) . Координати нової точки знаходять, знаючи кут φ , на який спрямований вектор OP (рис. 3, б), і визначаючи кут $\alpha = \widehat{POE^*}$ за теоремою косинусів:

$$W^2 = R^2 + \rho^2 - 2R\rho \cdot \cos(\alpha), \quad (1.6)$$

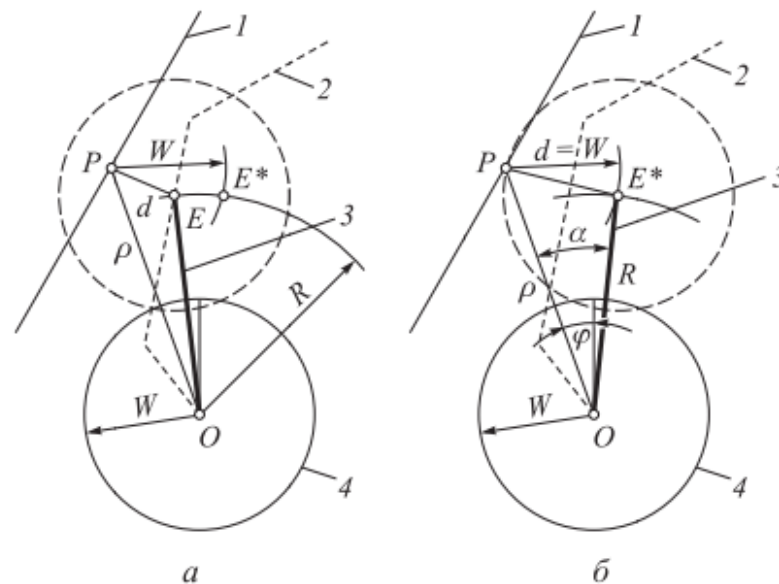


Рис. 1.4. Розрахункова схема алгоритму проходження бар'єру: а – до обробки; б – після корекції (1 – перешкода; 2 – заданий шлях; 3 – активний шлях; 4 – загальний оберт мобільного робота)

Зауважимо, що рівняння (1.6) має два розв'язки, які відрізняються знаком. Це перекриття відповідає положенню контрольної точки ліворуч і праворуч від перешкоди. Оптимальний розв'язок рівняння (1.6) має вигляд

$$\alpha = \arccos\left(\frac{R^2 + \rho^2 - W^2}{2R\rho}\right), \quad (1.7)$$

Координати нової точки відліку E^* відносно перешкод залежать від розташування старої точки відліку E . Знаючи вектори OE і OP , визначаємо орієнтацію ефективної траєкторії та знаходимо скоригований вектор OE^* :

$$OE^* = \begin{cases} (R\cos(\varphi - \alpha), R\sin(\varphi - \alpha)), OE \times OP > 0 \\ (R\cos(\varphi + \alpha), R\sin(\varphi + \alpha)), OE \times OP \leq 0 \end{cases} \quad (1,8)$$

Коли робот рухається, наступне опорне положення обчислюється та порівнюється з попереднім. З усіх точок вибирається найдальша від перешкоди. Оцінка відстані проводиться окремо для точок, розташованих зліва і справа від перешкоди. Зверніть увагу, що робот почне обертатися навколо перешкоди безпосередньо перед тим, як зіткнеться з нею.

Отже, модифікація траєкторії активного шляху дозволяє визначити координати поточної точки відліку E^* (рисунок 1.3) і перевірити рух робота по заданому шляху без зіткнення з нерухомими перешкодами.

Змінні бар'єри. Переходимо до завдання визначення координат точки відліку за наявності рухомих перешкод. Розрахунок параметрів динамічних перешкод, з одного боку, дозволяє уникнути зіткнення з ними, а з іншого – покращити траєкторію руху. Розглянемо простий приклад, показаний на малюнку 1.5. Нехай робот 1 рухається прямолінійно по деякій частині наміченого шляху. При цьому робот 2 (динамічна перешкода) рухається в конвергентному напрямку, стаючи перешкодою для першого.

Нехай задані параметри рухомого об'єкта: його положення в скінченній системі координат, загальні розміри, а також швидкість. Для вирішення поставленої задачі використовувався метод модифікованого ефективного методу, описаний вище. Тільки тепер, коли є рухомі перешкоди, розташування карти заповнення біля робота змінює її вміст відповідно до часу спостереження та параметрів рухомих об'єктів. У майбутньому така частина карти заповнення буде називатися динамічною.

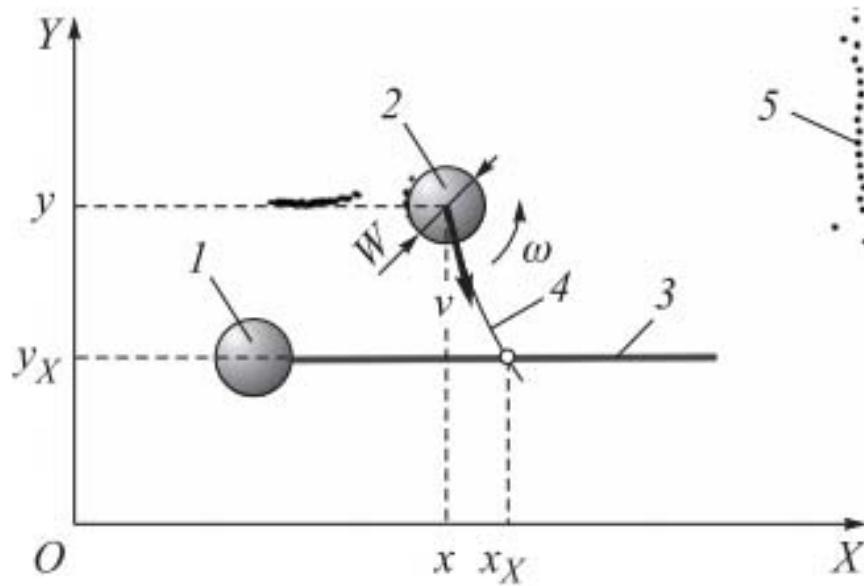


Рис. 1.5. Постановка динамічної задачі подолання перешкод: 1 – мобільний робот; 2 – змінна (обмеження); 3 – задана траєкторія руху робота; 4 – траєкторія руху змінного об'єкта; 5 – нерухомі шлагбауми

Щоб використовувати змінну, дії потрібно виконувати в такому порядку:

- 1) за параметрами рухомого об'єкта (перешкоди) в поточний момент часу обчислюємо його параметри в наступні моменти часу;
- 2) в змінній області карти очищаємо комірки, розташовані в зоні поточного стану об'єкта;
- 3) розрахувати місце та час перетину шляху перешкоди та шляху робота;
- 4) додати до карти динамічної категорії комірки, що визначають перешкоди на перетині траєкторій.

Після виконання перелічених дій ви можете знайти опорну точку за допомогою модифікованого методу активного шляху, замінивши результуючу секцію сили замість вихідної ділянки карти заповнення.

По-перше, розглянемо отримання значень параметрів об'єкта в реальному часі. У [5] запропоновано дві кінетичні моделі. Розглянемо одну з них – модель робота-перешкоди. На основі рисунка 1.5 вектор позиції об'єкта, що рухається в даний момент часу t_0 , визначається наступним чином :

$$s = (x \ y \ \varphi \ v \ \omega \ W)^T, \quad (1,9)$$

де x , y - положення об'єкта в скінченній системі координат; φ - кут повороту об'єкта відносно осі X скінченної системи координат; v , ω - швидкість руху і обертання об'єкта відповідно W ; R - радіус кола, визначеного навколо об'єкта.

Будемо вважати, що загальна маса W об'єкта постійна при разів, тобто предмет є твердим тілом фіксованої форми. Крім того, припустимо, що швидкість v , ω об'єкта постійна поблизу t_0 . Тоді значення решти векторних змінних (1.9) в довільний момент часу t можна розрахувати за допомогою рівнянь кінематики:

$$\varphi(t) = \int_{t_0}^t \omega dt + \varphi(t_0); \quad (1,10)$$

$$x(t) = \int_{t_0}^t v \cos \varphi(t) dt + x(t_0); \quad (1,11)$$

$$y(t) = \int_{t_0}^t v \sin \varphi(t) dt + y(t_0); \quad (1,12)$$

Зверніть увагу, що вирази (1.10 - 1.12) дійсні лише в регіоні біля $t_0 = (t \approx t_0)$

Наступний етап - очищення комірок області карти - це тривіальне завдання. Однак необхідно уточнити, що з цього моменту тоді алгоритм працює не з картою, що зберігається в пам'яті, а з копією сусіднього простору робота (за динамічною метою). У цьому випадку всім клітинкам у межах кола з центром у точці $(x(t_0); y(t_0))$ та радіусом W присвоюється статус «вільний». Таким чином, якщо предмет був старим додано на карту як статичний, а потім після цього процесу це область карти вважається вільною.

Для обчислення точки перетину використовується ітераційна процедура. Оскільки інтегрування рівнянь (1.10 - 1.12) виконується чисельно методу в кожній ітерації є відрізок мінімальної довжини, що визначає частину траєкторії об'єкта – перешкоду. як поточного шляху робота приймається заданий кусково-лінійний шлях у проблемній ситуації з припущенням, що відхилення фактичного руху робота від заданого шляху є незначним.

Потім для визначення точки перетину (x_X, y_X) в кожній ітерації вирішується задача про перетин поточної частини сумісної траєкторії перешкоди та штрихової лінії шляху робота. При цьому час перетину t_X визначається номером поточної ітерації , $i: t_X = t_0 + i \cdot \Delta t$, де t – крок інтегрування.

Останнім кроком є додавання клітинок до змінної області карти. Для цього клітинкам у межах кола з центром у точці X і радіусом W присвоюється статус «зайнято». Тепер для робота справжньою перешкодою є не щось у його поточному стані, а якийсь уявний об'єкт, який перебуватиме на перетині шляхів до певного моменту часу t_X . Іншими словами, метод ефективного шляху за допомогою виразів (1.2) і (1.3) обчислює точку відліку з урахуванням координат майбутнього положення рухомого об'єкта (перешкоди).

Для оцінки ефективності методу вирішення задачі керування рухом мобільного робота за наявності динамічних перешкод було проведено моделювання. На малюнку 1.6 показано два роботи. Правий робот рухається по заданому шляху (по прямій). Зліва знаходиться другий робот - динамічний об'єкт, який обмежує рух першого робота. Він рухається по криволінійній траєкторії. Під час симуляції обидва роботи починають рухатися одночасно.

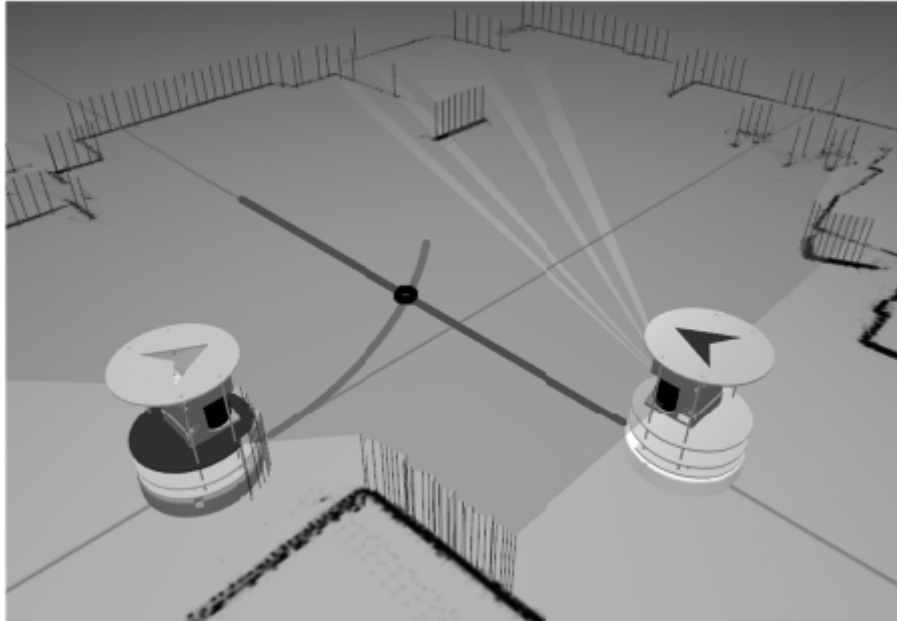


Рис. 1.6. Моделювання процесу управління

Розроблено два варіанти моделювання: перший - з використанням алгоритму обходу фіксованої перешкоди, другий - з використанням алгоритму динамічного обходу перешкоди. Як видно на рисунку 1.7, у першому випадку робот розганявся до великої швидкості. Потім на шляху виникла перешкода, і робот почав повертатися, намагаючись її обійти. Однак рух почався надто пізно, і в районі сталося зіткнення t_x .

У другому випадку (рис. 1.7, б) робот почав розганятися, але як тільки в точці перетину траєкторій з'являлася уявна перешкода, лінійна швидкість сповільнювалася. При цьому відбувалися коливання кутової швидкості. Це означає, що точка відліку була отримана в кожній точці часу з різних боків перешкоди, таким чином зменшуючи лінійну швидкість руху. Після того, як перешкода зникла, робот продовжив рух до мети на великій швидкості і досяг її вчасно t_f .

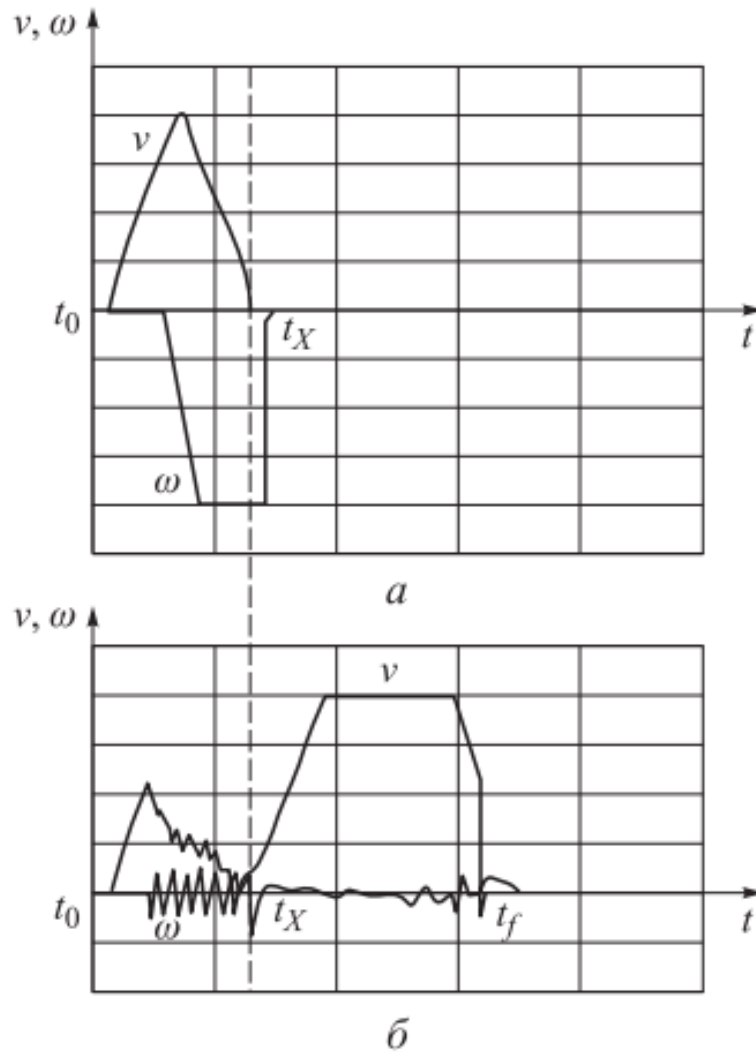


Рис. 1. 7 . Симуляція є успішною під час роботи у фіксованому середовищі (а) та за наявності динамічних перешкод (б).

Потім досліджувалася задача керування рухом мобільного робота по заданій траєкторії за наявності динамічних перешкод. Як рішення пропонується модифікація активного методу. Результати моделювання підтвердили ефективність методу. Координати опорних точок, отримані в режимі реального часу, можуть бути використані в системі керування мобільним роботом для формування шляху в обхід перешкод.

1. 3. Аналіз методів та алгоритмів визначення параметрів об'єкта та відстані до нього на основі обробки зображень

Системи поділяються за принципом використання технічного обладнання, що дозволяє отримувати зображення:

- одна камера і вимірюваний об'єкт або розмічальний конус;
- оптичні системи;
- одна камера і лазерний далекомір;
- дві камери (принцип стереометрії).

Використовуючи ту саму камеру та вимірювальний об'єкт або маркувальний конус.

Система використовує зображення з відеокамери, на якому, крім об'єкта, має бути присутнім вимірювальна лінійка (рулетка) (рисунок 1.8). Зображення, отримане в камеру, показує предмет і лінійку прямо в центрі зображення. Зображення калібрується за допомогою лінійки, а потім координати об'єкта на зображенні порівнюються з позначкою на лінійці.

Для калібрування зображення достатньо знати розмір лінійки в пікселях і метрах, а також координату пікселя, який є фактичним центром лінійки.

Для розрахунку необхідної відстані використовується наступна формула:

$$I = \frac{(L \times K)}{\frac{W}{x} - 1 + k}, \quad (1.13)$$

де I досліджувана відстань до об'єкта, м; L - висота лінійки, м; W - висота лінійки в пікселях; x - перекриття об'єкта на зображенні;

$$K = \frac{W - M}{M}, \quad (1.14)$$

коефіцієнт, що вказує кут нахилу камери, де M – координата центру лінійки.

На малюнку 1.8. передбачено вікно програми для визначення відстані до об'єкта та його швидкості.

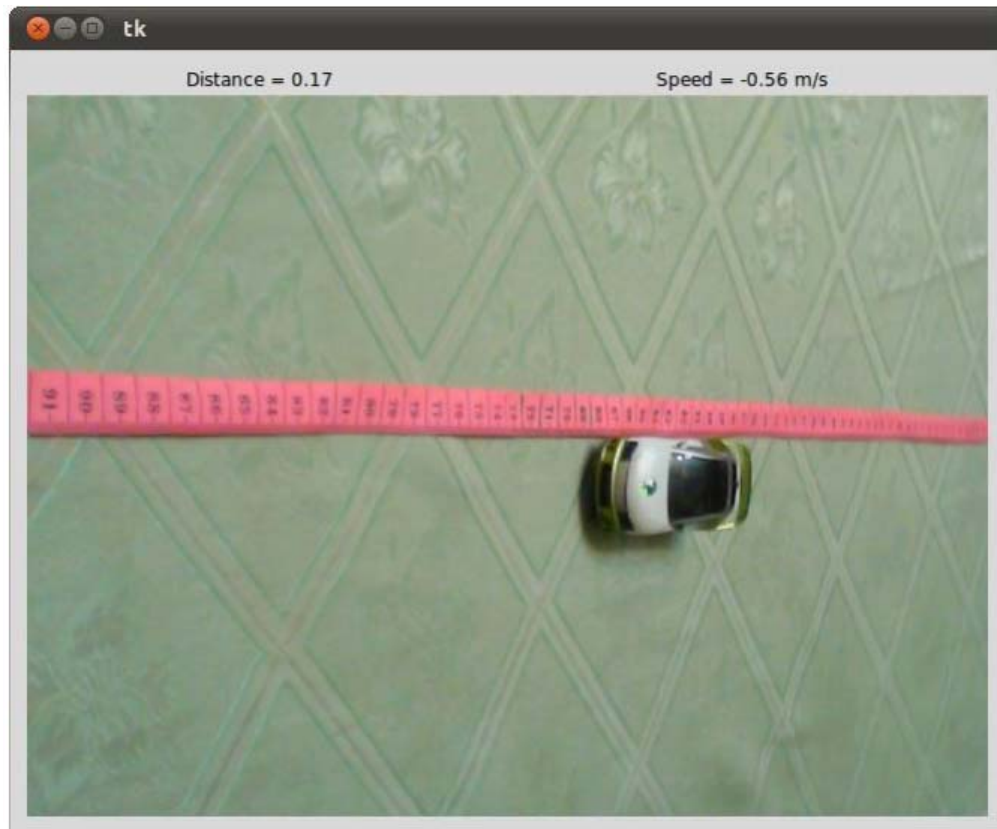


Рис. 1.8. Програма для визначення відстані до об'єкта та його швидкості.

Реалізація методу представлена мовою Python з використанням бібліотеки OpenCV. Замість лінійки можна використовувати розмічальний конус, розташований у центрі вимірюваної площі. Подібні методи будуть добре працювати при вимірюванні розмірів у статичних моделях і системах.

Застосування оптичних систем. Оптичні методи — це застосування принципів оптики для отримання зображень різного масштабу в одному об'єкті.

Один з алгоритмів визначення розташування і розмірів об'єктів передбачає послідовне вирівнювання і віднімання екрану або лінзи відносно один одного вздовж оптичної осі в деякому заданому діапазоні Δx (рис. 1.9).

Як оптичну систему можна використати більш складну систему лінз або зображення буде побудовано за схемою, наведеною на рисунку 1.10.

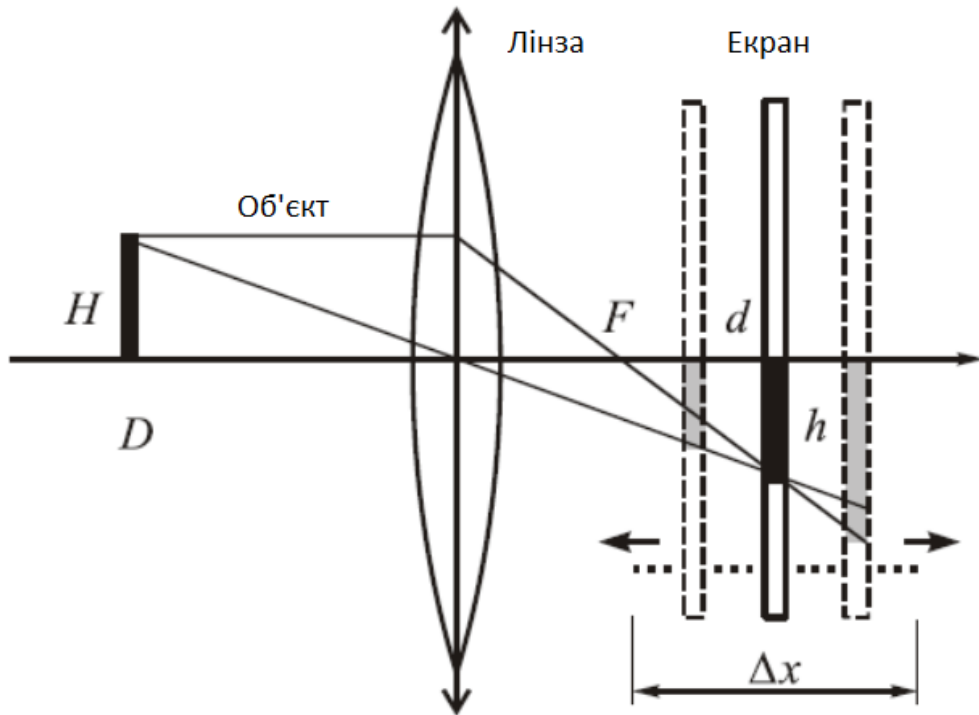


Рис. 1.9. Формування зображення на рухомому екрані за допомогою однієї лінзи

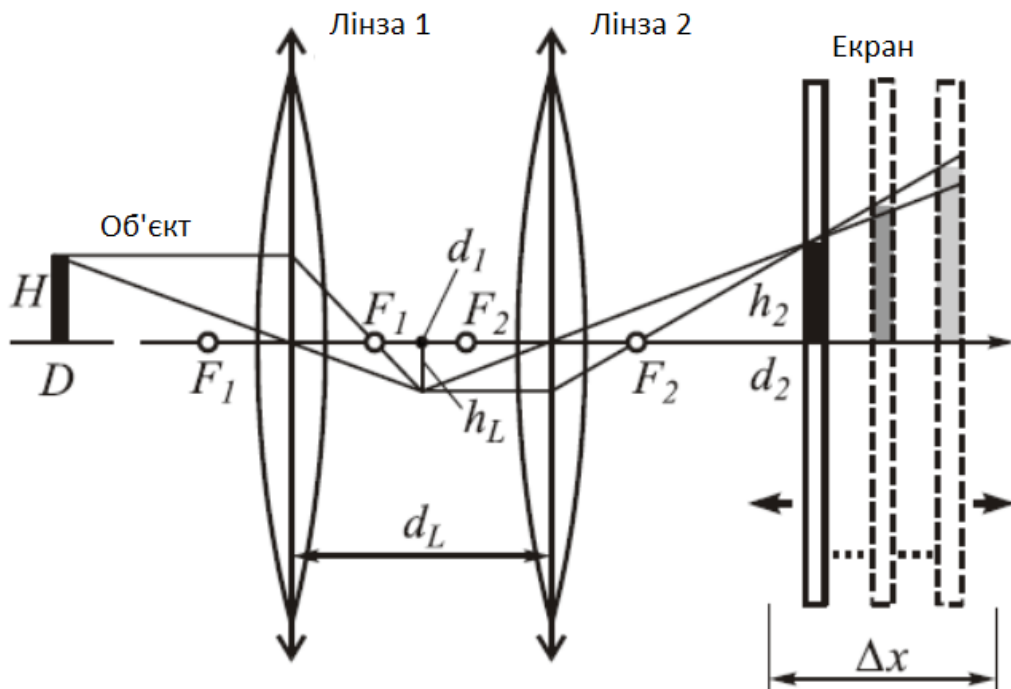


Рис. 1.10. Формування зображення на рухомому екрані за допомогою двох лінз

У результаті відстань позиції 1 (рисунок 1.9) визначається за такою формулою

$$D_i = \frac{d_i \times F}{d_i - F}, \quad (1,15)$$

де

D_i - відстань від предмета до лінзи i -го зображення масиву; F - фокусна відстань об'єктива; d_i - відстань від лінзи до зображення i -го кадру масиву.

В іншому випадку (рисунок 1.10) значення D визначається шляхом застосування формули 1.13 по черзі для кожної лінзи.

Інший метод полягає у визначенні відстані до об'єкта через різні інтервали часу за допомогою оптичного фотоприймача з регульованою оптичною системою з двома відомими фокусними відстанями (рисунок 1.11) [3]. В результаті отримують два зображення різного масштабу і визначають відстань до об'єкта за такою формулою

$$a = f_2' * \frac{1 - \frac{Y_1'}{Y_2'}}{\frac{Y_1' * f_1' - 1}{Y_2' * f_2' - 1}}, \quad (1,16)$$

де

f_1 а f_2 - максимальна фокусна відстань оптичної системи; Y_1' і Y_2' , - розмір зображень об'єкта в f_1 і відповідно f_2 .

Ці методи мають високу точність показаних результатів, але наявність механічного руху в процесі отримання зображення впливає на швидкість отримання результатів, як наслідок, такі системи будуть непрактичними, коли швидкі алгоритми застосовують більші роботизовані рухи.

З використанням однієї камери та лазерного далекоміра.

Використання комплексу відеокамер і лазерного далекоміра дозволяє вимірювати відстань до об'єкта, бінарне зображення якого виходить в результаті застосування алгоритму розпізнавання форми (рисунок 1.12).

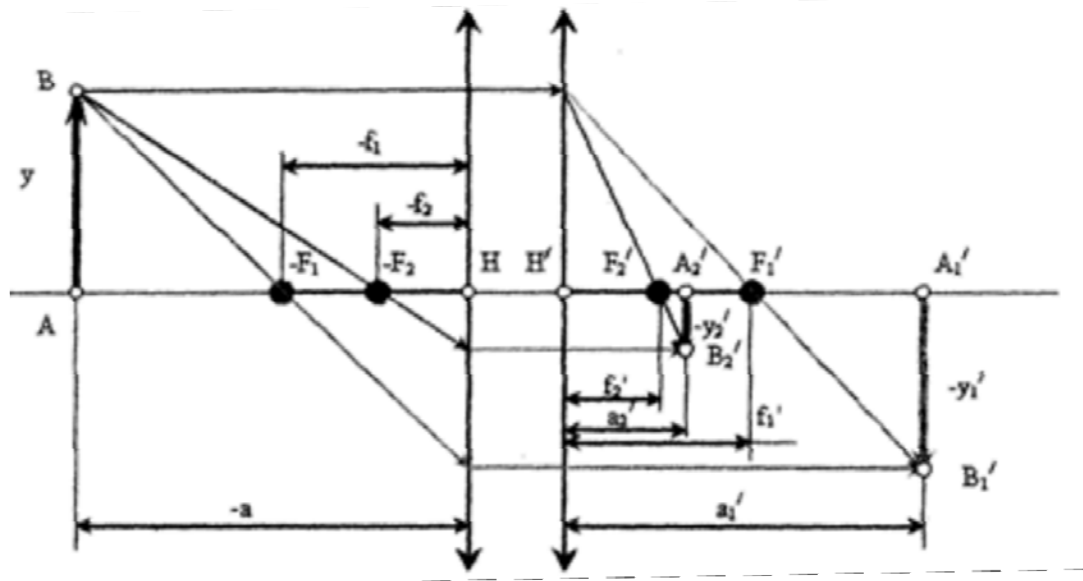


Рис. 1.11. Взаємне розташування предмета і його зображення при максимальній фокусній відстані f_1 і f_2

Після бінаризації зображення відстань до об'єкта вимірюється лазерним далекоміром, встановленим на поворотній осі.

Запропонований алгоритм може гарантувати достатню точність вимірювання, але наявність механічного руху, як і в попередніх методах, істотно впливає на час отримання результату.

Використання двох камер (принцип стереометрії). Алгоритми визначення відстані, які використовують дві камери, припускають, що камери знаходяться на певній відстані одна від одної та на одній осі щодо досліджуваного об'єкта, а осі їх оптичних – перпендикулярні положенню об'єкта.

Так, наприклад, один із алгоритмів використовує дві USB-камери, і, отже, стереоскопічне зображення, побудова Z-зображення або зображення глибини є одноетапним зображенням, значення кожного пікселя його $w_a p_a$ дорівнює відстань до візуального об'єкта [5].

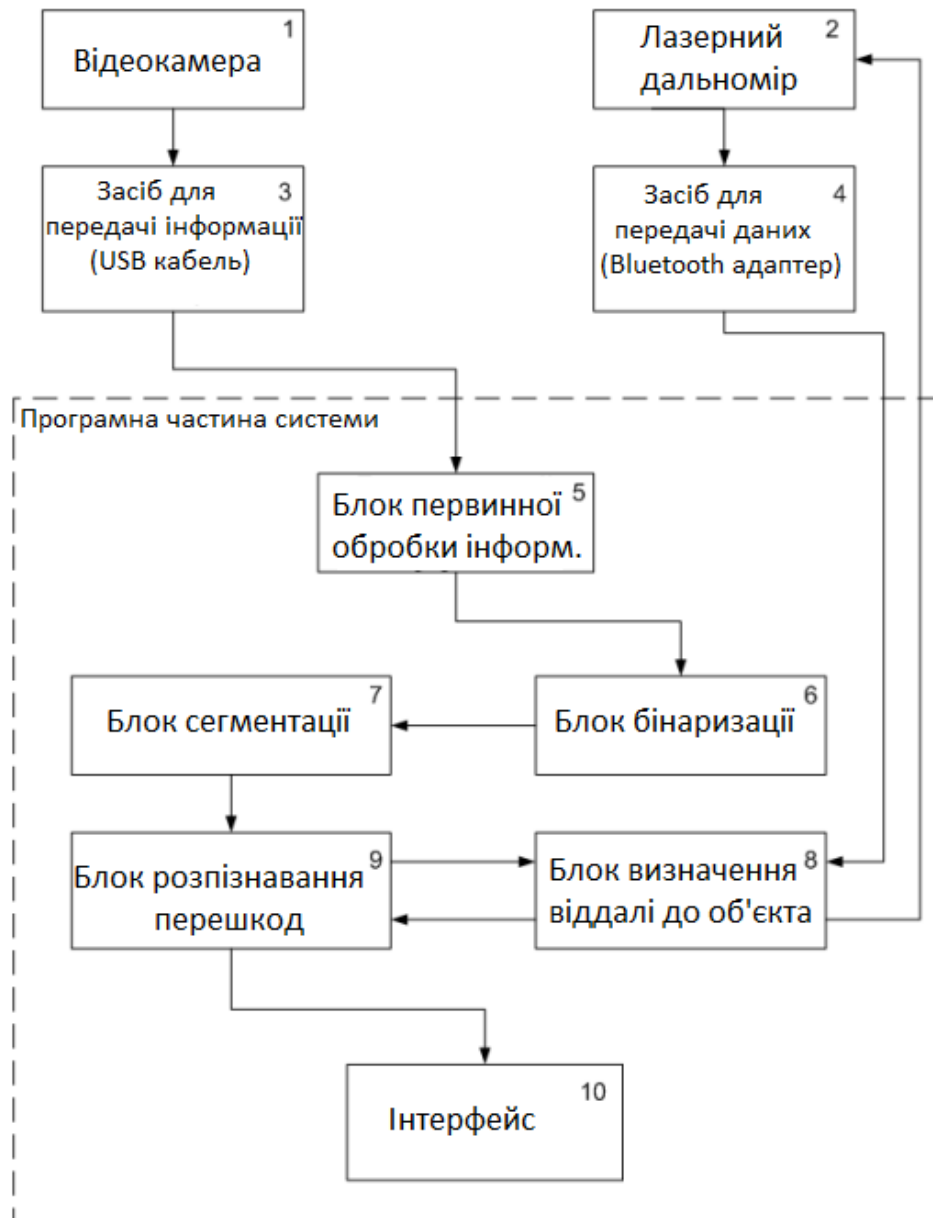


Рис. 1.12. Схема алгоритму розпізнавання форми та визначення відстані до об'єкта

Суть процедури полягає в наступному: для кожної точки на одному зображенні шукають відповідну точку на іншому зображенні; можна провести триангуляцію на парі відповідних точок і визначити їх прототипні координати в тривимірному просторі; знаючи тривимірні координати зразка, глибина розраховується як відстань до площини камери. Для кожного пікселя лівого зображення з координатами (x_0, y_0) виконується пошук пікселів у правому зображенні. Передбачається, що піксель правого зображення повинен мати

координати $(x_0 - d, y_0)$, де d - зміщення камер відносно одна одної. У результаті виходить зображення глибини.

Інший запропонований метод вимірювання відстані до об'єкта базується на принципах фотограмметрії та кореляційної обробки стереопарних цифрових зображень. Отримання інформації про дальність передбачає реєстрацію кількох зображень об'єктів під різними ракурсами.

Якщо відстань між камерою і точкою огляду значно перевищує фокусну відстань оптичної системи, можна вважати, що зображення побудовано в її фокальній площині. У моделі проєктивної камери (рисунок 1.13) зображення тривимірного об'єкта отримують проєктуванням його на фокальну площину.

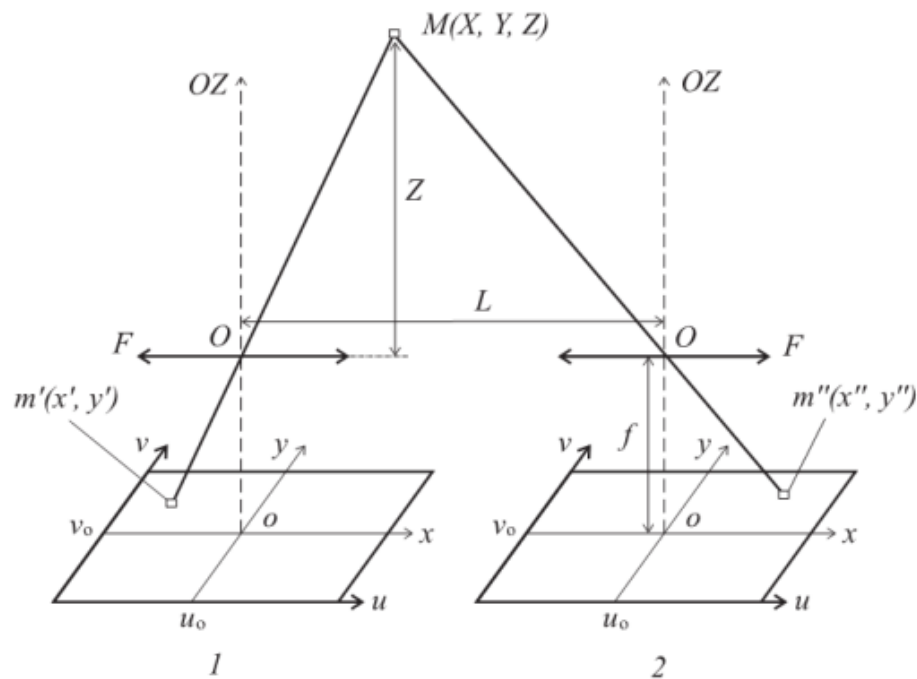


Рис. 1.13. Проекційна модель камери і стереосистеми: f - фокусна відстань, L - основна висота, Z - відстань до об'єкта.

Проєкцією точки в тривимірному просторі M з координатами (X, Y, Z) , де Z відстань, є точка m з координатами (x, y) , при цьому виконується таке співвідношення:

$$x = fX/Z, y = fY/Z \quad (1.17)$$

де

f - фокусна відстань оптичної системи (фотоапарата).

В результаті багатьох обчислень вони приходять до того, що відстань до об'єкта Z становить геометрія зображень визначається співвідношенням

$$Z = f * \frac{L}{x' - x''}, \quad (1,18)$$

де

f - фокусна відстань, L - основний розмір (відстань між оптичними осями камер), .

x' і x'' - координата об'єкта в площині зображення на правому і лівому зображеннях стереопари відповідно, а їх відмінність – зміщення об'єкта в площині зображення на першому та другому зображеннях стереопари відповідно.

Розглянуті два методи забезпечують більшу точність вимірювань, недолік механічних частини та інструменти оперативного радіолокаційного принципу дозволяють швидко і точно розраховувати відстань до об'єкта та інші параметри огляду.

1. 4 . Аналіз методів визначення відстані до перешкод під час руху мобільного робота

Автономні мобільні пристрої стають все більш важливими в повсякденному житті людини. Серед таких застосувань особливу нішу займають мобільні роботи: прибиральники, обслуговуючі працівники, екскурсоводи, домашні помічники, навігаційні пристрої та інші методи, призначені для виконання різноманітних завдань у домашніх умовах, промислових, аварійних та інших, оснащених інноваційною людиною-

машиною. інтерфейси. Характеристикою продуктивності мобільних роботів є необхідність перевірки їхньої роботи в складних, слабороздільних середовищах: у різних приміщеннях і на відкритому просторі. Щоб забезпечити орієнтацію в такому середовищі, мобільний робот повинен бути оснащений розробленою системою орієнтації, одним із компонентів якої є система технічного зору.

При застосуванні систем технічного зору для мобільних роботів, що працюють у невідомому або непередбачуваному середовищі, одним із найважливіших завдань є розпізнавання перешкод, тобто об'єктів на шляху їх руху. Вирішення цього завдання є дуже складним процесом, оскільки він тісно пов'язаний з методами попередньої обробки зображень. Часто якість розпізнавання залежить від ефективності ряду процесів, починаючи з проникнення в зображення перешкоди і закінчуючи формулюванням його ознак. Найважливішим завданням розпізнавання є визначення ознак перешкоди в процесі її ідентифікації.

Ідентифікація перешкод передбачає об'єднання всієї отриманої інформації в один об'єкт виділення перешкоди. Більшість відомих методів не проводять чіткої межі між процесами розпізнавання та ідентифікації, оскільки виділення ознак відбувається за рахунок модифікації вихідного зображення [11].

Найбільш перспективною та перспективною сферою застосування систем технічного зору є зорове сприйняття мобільних роботів. Необхідність вільного пересування в певному середовищі значно підвищує вимоги до швидкісних характеристик систем технічного зору, оскільки вони повинні забезпечувати ідентифікацію та розпізнавання об'єктів у навколишньому середовищі з мінімальними витратами часу та меншими витратами енергії.

Для забезпечення орієнтації робота в навколишньому середовищі використовуються переважно різні далекоміри: ультразвукові, оптичні та лазерні [12]. Ultrasonics характеризується великим часом відгуку - близько десяти секунд, і якщо робот знаходиться у великому відкритому просторі, це

не дозволяє швидко рухатися. Швидкість звуку в різних налаштуваннях також може змінюватися, впливаючи на точність оцінки відстані, що призводить до спотворення загальної картини навколишнього середовища. За допомогою лазерних далекомірів можна отримати зображення навколишнього середовища тільки в точній зоні зору. Крім того, на шляху променя часто є невеликі перешкоди, які вносять похибку в цьому напрямку. Створювати тривимірні карти за допомогою лазерів у масштабі реального часу дуже складно і, як мінімум, вимагає великої обчислювальної потужності.

Час до перешкоди також визначається за допомогою двох відеокамер [12], що дозволяє працювати над побудовою 3D моделі перешкоди, але потребує більше часу для аналізу отриманих даних. Об'єднання обох зображень в одне вимагає складного та енергоємного алгоритму. Якщо відеокамери розташовані складним чином, то на основі відомої відстані між ними можна об'єднати зображення. Але якщо на шляху будь-якої є одна перешкода, а друга одночасно зареєструє іншу перешкоду, то виникає проблема з'єднання опорних точок.

У цьому випадку знайти друга – завдання непосильна. Для їх вирішення потрібно буде знайти збіги не тільки опорних точок зображення, а й прямих ліній або фігур. Головним недоліком такого підходу є висока вартість технічних прийомів.

Існує алгоритм визначення відстані до перешкоди за допомогою однієї відеокамери на рівній поверхні, який дозволяє оцінити відстань до перешкод та їх напрямок в одному чорно-білому кадрі, згідно з наступними припущеннями [13]. : 1 .

- виконано початкове калібрування камери;
- камера фіксується в нерухомому режимі, .
- робот рухається по рівній горизонтальній поверхні;
- в навколишньому середовищі відсутні зважені частки.

Однак ці припущення не відповідають умовам забезпечення автономного руху мобільного робота в частково або повністю невизначеному середовищі, для вирішення яких необхідно:

- розпізнавання та визначення параметрів перешкод, з якими стикається працівник під час його руху;
- задати напрямок (траєкторію) руху робота.

По-перше, необхідно вирішити проблему розпізнавання, оскільки від її вирішення залежить ефективність маршрутизації.

Для розпізнавання та визначення параметрів перешкоди можна використовувати одну відеокамеру та алгоритм розпізнавання перешкод, який є частиною системи керування мобільним роботом і контролює робочий простір перед роботом за траєкторією, по якій він рухається.

Алгоритм складається з наступних кроків:

- отримання інтерпретації зображення навколишнього середовища з відеокамери;
- обробка зображень;
- усвідомлення бар'єрів;
- визначення параметрів і орієнтування перешкоди;

Спосіб визначення параметрів і орієнтації перешкоди полягає в наступному.

Відеокамера має два відомі параметри: ширину кадру L у пікселях, ρ_x , і кут огляду камери Ω , град. Коли виникає перешкода, робот визначає ширину перешкоди l у пікселях. Кутовий розмір перешкоди γ , град. (рисунок 1.14), виходить із рівнянь

$$\frac{\Omega}{L} = \frac{\gamma}{l}, \gamma = \frac{\Omega l}{L}, \quad (1.19)$$

Після виконання цього розрахунку робот рухається вперед на певну відстань і повторює обчислення. Порівнюючи отримані дані, можна визначити фактичну відстань до перешкоди (рисунок 1.15) за такою системою рівнянь:

$$\begin{cases} \operatorname{tg}\alpha = \frac{x}{y+a} \\ \operatorname{tg}\beta = \frac{x}{y} \end{cases}, \quad (1,20)$$

де

x - половина ширини шлагбаума;

y - відстань до перешкоди;

α - половина кутового розміру перешкоди при першому вимірюванні;

β - половина кутового розміру перешкоди при другому вимірюванні;

a - відстань, пройдену роботом між зарядами.

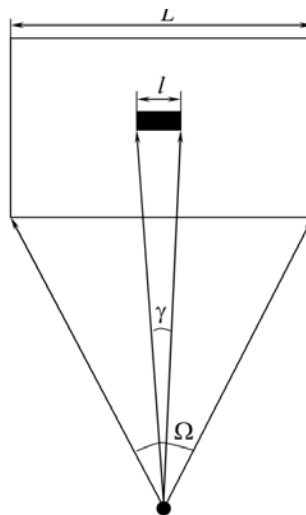


Рис. 1.14. Розрахунок кутової величини об'єкта

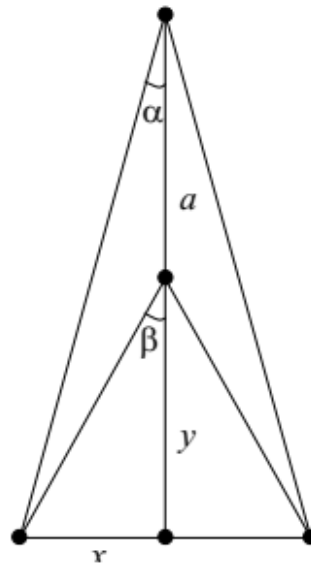


Рис. 1.15. Графічне зображення системи рівнянь

Цей спосіб визначення відстані до перешкоди під час руху мобільного робота, на відміну від більшості існуючих, не є жорстко прив'язаним до конкретних умов руху та додає гнучкості та автономності системи регулювання в цілому. Алгоритм не потребує великої обчислювальної потужності та економічно дуже вигідний. Він також має низьке енергоспоживання, що є не менш важливою характеристикою для рухової системи мобільного робота.

1.5. Постановка проблеми дослідження

Технічний прогрес у робототехніці, автомобілебудуванні та інших галузях промисловості досяг такого рівня, що системи комп'ютерного бачення, і особливо стереобачення, широко поширені та швидко розвиваються.

Це сталося у зв'язку зі зростанням обчислювальної потужності і розширеними завданнями, які постають перед робототехнічними додатками, адже за допомогою «бачіння» машини можуть отримувати велику кількість інформації про навколишнє середовище.

Метою даної роботи є розробка алгоритму визначення відстані до перешкод мобільного робота на основі обробки стереозображення.

Для досягнення поставленої мети необхідно вирішити основні актуальні завдання:

- аналіз проблем використання комп'ютерного зору, визначення основних напрямів їх удосконалення;
 - проаналізувати з точки зору керування рухом мобільного робота та провести класифікацію потенційних перешкод, що виникають у процесі руху мобільного робота.
 - дослідження методів і алгоритмів визначення параметрів об'єкта та відстані до нього на основі обробки зображень;
 - аналіз методів визначення відстані до перешкод під час руху мобільного робота;
 - розробка алгоритму розрахунку відстані до спостережуваного об'єкта за зображеннями зі стереопари;
 - побудова схем у стереомоделі реалізації обраного алгоритму;
 - процедури калібрування, спотворення та обробки зображень;
 - побудова карти глибини та побудова математичної моделі формування стереопотоку;
 - виконати програмну реалізацію системи визначення відстані до перешкод мобільного робота;
- проведення експериментальних досліджень і розробка рекомендацій щодо практичного застосування розроблених методів і алгоритмів.

Підсумок першого розділу

1. Проведено аналіз проблем використання комп'ютерного зору, визначено основні фактори, що формують їх вдосконалення.

2. Розроблено пропозицію щодо керування рухом мобільного робота та розраховано потенційні перешкоди , які виникають під час руху мобільного робота .

3. Проведено аналіз методів і алгоритмів визначення параметрів об'єкта та відстані до нього на основі обробки зображень, виділено основні проблемні моменти в процесі їх реалізації.

4. Проведено аналіз методів визначення відстані до перешкод під час руху мобільного робота.

РОЗДІЛ

РОЗГЛЯД ДЛЯ ВИЯВЛЕННЯ СПЕЦИФІКАЦІЙ МОБІЛЬНОГО РОБОТА

2.1.1. Як визначити відстань до перешкод

У рамках магістерської роботи запропоновано алгоритм розрахунку відстані до спостережуваного об'єкта за зображеннями зі стереопари. Визначення відстані до об'єкта за зображеннями з двох камер є однією з ключових функцій систем комп'ютерного зору. У лабораторіях світу ведуться дослідження практичного застосування стереозору для отримання інформації про навколишній світ.

Одним із застосувань стереозору є пересування мобільних роботів. За допомогою системи стереобачення робот може визначити відстань до перешкоди та успішно її пройти.

Під час майстер-класу використовувалися наступні інструменти:

- дві USB камери;
- шаховий розмір 594 x 841 мм;
- операційною системою Linux , Python 2.7;
- мірна стрічка.

По-перше, робота була виконана до розробки алгоритму:

- створено шаблон шахової дошки розміром 594 x 841 мм на основі спеціального шаблону з відкритої бібліотеки OpenCV. Це зображення має 6 вертикальних вузлів і 9 горизонтальних вузлів.

2. Створення підставки з двох USB-камер, розміщених на рівній твердій поверхні, розташованих на фіксованій відстані одна від одної.

у обчислити відстань до об'єкта спостереження. Давайте розглянемо більш детальний опис кожного кроку цього алгоритму.

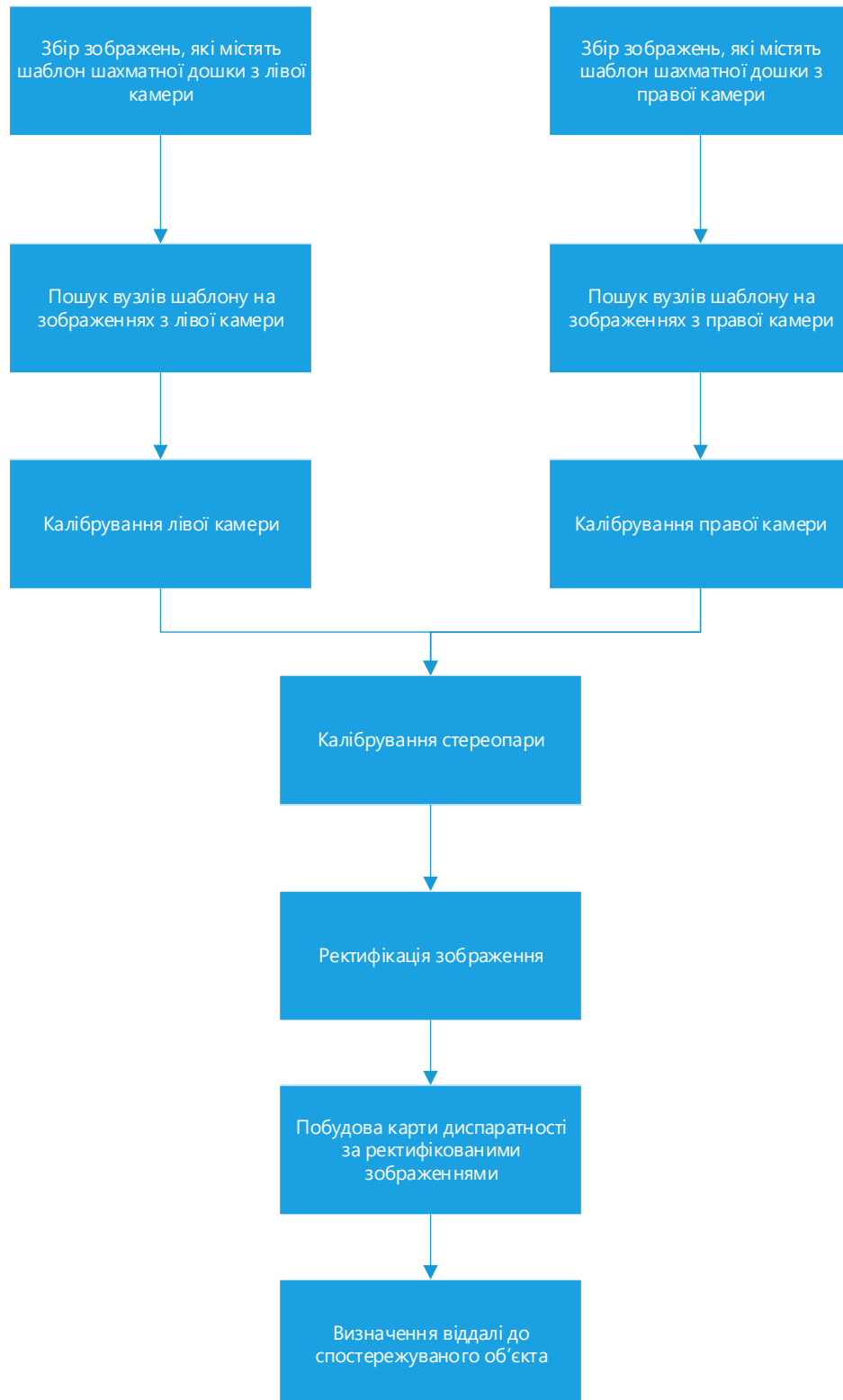


Рис. 2.1.1. Алгоритм визначення відстані до об'єкта

Крок 1. Першим кроком є збір зображень, які показують шаховий порядок у різних положеннях з лівої та правої камер стереопари. Для коректної

реалізації алгоритму необхідно, щоб шахове зображення повністю потрапляло в поле зору стереопарних камер і займало якомога більшу площу кадру.

Крок 2. На другому кроці здійснюється пошук вузлів шахової дошки на зображеннях, отриманих на кроці 1. Пошук вузлів шаблонів здійснюється за допомогою бібліотечної функції OpenCV `findChessboardCorners()`. Координати вузлів зображення зберігаються в іншій програмі.

Крок 3. На третьому кроці кожна зі стереопар калібрується окремо. Стереопарні камери калібруються за допомогою бібліотечної функції OpenCV `CalibrateCamera()`.

Крок 4. На четвертому кроці відбувається калібрування всієї стереопари. Дані, отримані на третьому кроці, надходять на вхід бібліотечної функції OpenCV `stereoCalibrate()`, яка обчислює внутрішні параметри стереопари. Після процесу стереокалібрування ми отримуємо фокусні відстані камер у пікселях і координати головних точок камер.

Крок 5. На п'ятому етапі знімають спотворення зображень, отриманих на першому кроці. Викривлення усуваються за допомогою функцій бібліотеки OpenCV `initUndistortRectifyMap()` і `remap()`. Після виконання цього кроку всі перпендикулярні лінії спостережень у навколишньому світі стають перпендикулярними лініями на зображенні.

Крок 6. На шостому етапі зображення, отримані на кроці 5, коригуються. Корекція полягає у вирівнюванні деформованих зображень, отриманих на кроці 5, таким чином, щоб епіполлярні лінії зображень з лівої камери збігалися з епіполлярними лініями зображень з відповідної камери. Цей процес виконується бібліотечною функцією OpenCV `stereoRectify()`.

Крок 7. На сьомому кроці для кожної пари відповідних зображень із лівої та правої камер створюється контрастна карта (або карта глибини) на основі зображень, отриманих на шостому кроці. Кожен піксель карти контрасту містить інформацію про те, скільки пікселів уздовж осі OX лежить між відповідними пікселями зображень об'єкта з лівої та правої камер. Побудова

різницевої карти базується на алгоритмі SGBM з використанням функції бібліотеки OpenCV Stereo SGBM.

Крок 8. На шостому етапі обчислюється відстань до спостережуваного об'єкта. Розрахунок відстані до об'єкта безпосередньо відповідає інформації, що міститься на карті різниць, побудованій на сьомому кроці.

Розглянемо детально формули для знаходження відстані до спостережуваного об'єкта.

Нехай задані такі параметри:

1. відстань між камерами стереопари (базова відстань, базова довжина);
2. фокусна відстань пари стереокамер дорівнює f ;
3. Мепе різниця Знайти: відстань Z до спостережуваного об'єкта.

Розглянемо рішення більш детально:

P - щось спостерігав : .

$base$ - відстань між стереопарами камер (в метрах) ;

f - фокусна відстань стереопари камери (у пікселях) ;

x_0 - координата по осі ОХ точки об'єкта P на зображенні з лівої камери

x

$x_0 - d$ - координата по осі ОХ точки об'єкта P на зображенні з правої

камери

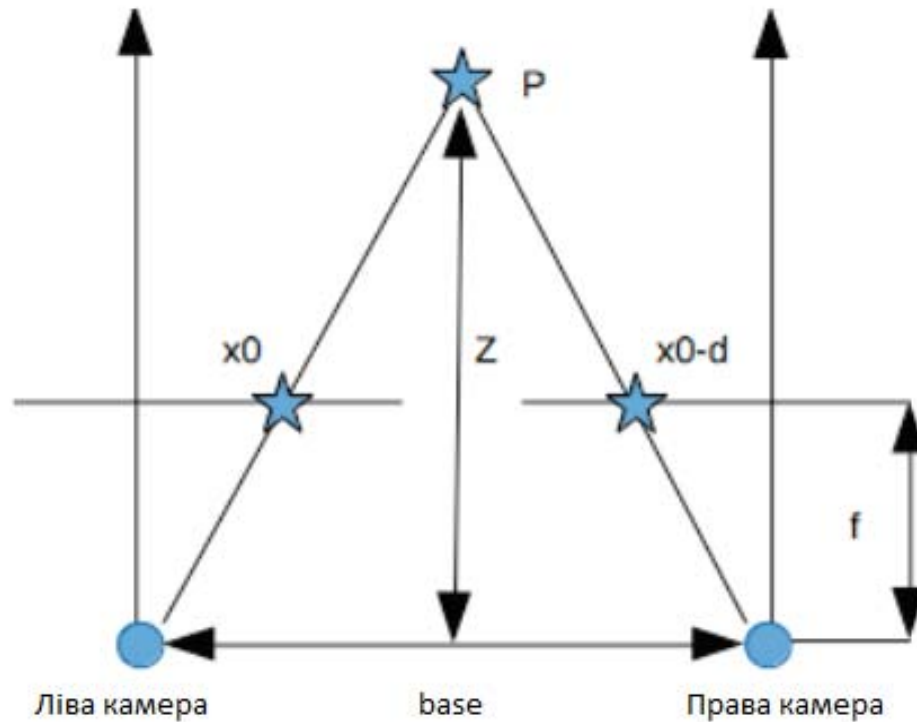


Рис. 2.2. Схема реалізації обчислювального процесу

f інтервал між заголовками x_0 елементів P (у пікселях);

Z - відстань до об'єкта спостереження (у метрах);

Виходячи з інформації, наведеної на малюнку 2.2, ви можете побачити подібні трикутники. Для подібних трикутників виконуються такі співвідношення:

$$(base - d)/(Z - f) = base/Z \quad (2.1)$$

З цього стандарту випливає наступне

$$Z = f * base/d \quad (2.2)$$

Ми знайшли формулу для обчислення відстані до спостережуваного об'єкта .

Вивчивши цю формулу, можна зробити наступні висновки:

1. Відстань до положення об'єкта обернено пропорційна переміщенню даної точки. Це означає, що точки, розташовані далі від площини стереопари, мають меншу дисперсію, ніж точки, розташовані ближче до площини стереопари.

2. Коефіцієнт контрастності прямо пропорційний відстані між камерами. Чим більша відстань між камерами, тим більший контраст.

3. Розширення карти контрасту лінійно залежить від розширення зображення. Чим більше збільшення зображення, тим точніша карта контрасту.

2.2. Стереомодель обраного алгоритму та збіг потоку

Розглянемо більш детально реалізацію стереорекострукції за зображеннями, отриманими з двох камер. Застосування концепції глибини до стереозображень базується на тому факті, що зображення будуються відповідно до тих самих геометричних принципів, що й зображення, коли воно потрапляє на сітківку ока. Два зображення одного і того ж об'єкта в один і той же час називають стереопарою. На рисунку 2.3 зображено стереомодель обраного алгоритму.

те ж саме, зображення були зроблені $P_{ЛВ}$ в двох місцях $S_{Лі}$ $S_{П}$ камерами $P_{ПЗ}$ однаковою фокусною відстанню f . Якщо отримані зображення розміщуються перед очима людини на тій самій відстані, що й об'єкти під час стрільби, то точки зображення, які є точковими зображеннями M_i N представлені на сітківці очей як m, n, m', n' _

Різні віддалені точки M_i N взаємні причини переміщень їх відображень на малюнках, ці переміщення (m', n') відповідно стають паралаксами S , на розмір, за якого мозок створює тривимірне зображення об'єкта.

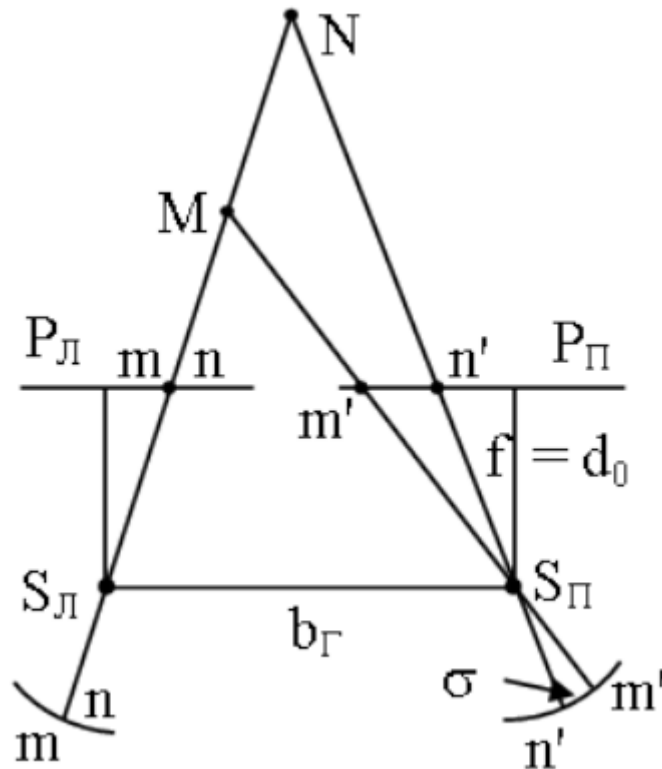


Рис. 2.3. Схема реалізації стереомоделі

Дуже важко оцінити фактичні умови зображення в параметрах людського зору, як наслідок, стереомодель не збігається з об'єктом зображення. Його горизонтальний масштаб $1:M$ може відрізнитися від вертикального $1:M_b$.

Це значення масштабу можна розрахувати за допомогою формули

$$\frac{M}{M_b} = \frac{250}{f} \quad (2.3)$$

Наприклад, при фокусній відстані $f = 100$ мм стереомодель розтягнеться вгору в 2,5 рази, при $f = 350$ мм стиснеться в 0,7 рази. Зміна збільшення спостережуваної системи не змінює цей факт, оскільки одночасно змінюється горизонтальний і вертикальний масштаби.

Використання звичайних прозорих камер для стереозображення викликає кілька проблем. Оптичне калібрування таких камер вимагає надзвичайно високої точності технології виготовлення пристрою. Щоб потоки

зображень, сформовані двома окремими камерами, називалися стереопотоком, вони повинні бути оброблені програмним забезпеченням. Для цього ви хочете:

- сумісність камери;
- калібрування кожної камери для визначення внутрішніх параметрів;
- стереокалібрування камер (калібрування до стандартної системи координат);
- прибрати спотворення;
- обробка зображень;

У цьому розділі ми продовжимо розглядати основні проблеми, які виникають, коли ми виходимо за межі вищезгаданих етапів, ми досліджуватимемо методи синхронізації відеопотоків, обробки зображень і усунення спотворень із зображень.

Матриця зображення $\mu_{W,H}(i,j)$ розміру зображення $W \times H$ в площині I отримується шляхом визначення термінів i і стовпців j зображення, а зображення $img_{W,H}(i,j)$ визначається як набір I різних функцій у матриці $\mu_{W,H}(i,j)$:

$$img_{W,H} \in I^{def} = \{\mu_{W,H} \rightarrow color\} \quad (2,4)$$

S - фрейму за допомогою набору фотографії та періоди часу, в які вони були зроблені

$$S^{def} = (img, r) \in I \times R \quad (2,5)$$

Тоді H (будемо розуміти під потоком кадрів). послідовність кадрів із збільшенням часу період:

$$F(i) \stackrel{def}{=} S(i), \forall (i \in Z_+) [F(i+1). \tau > F(i). \tau] \quad (2.6)$$

Відповідність функції узгодження потоку $F1(k), F2(n)$ із заданою специфічністю, визначеною як вибірковість прямо паралельні субпоточи кадрів δ :

$$\text{Synchronize}(F1(k), F2(n), \delta) \stackrel{\text{def}}{=} F1(ki)F2(ni) \quad (2.7) _$$

$$\text{Sync}(F1(ki), F2(ni), \delta) \stackrel{\text{def}}{=} F1(ki)F2(ni) \quad (2.8) _$$

Перш ніж ми підемо далі, ми отримаємо три частини координати точок, деякі необхідно проаналізувати геометричні фігури, які допомагають інтегрувати ієрархію точкових проєкцій у тривимірний простір.

Нехай є дві камери, як це видно на малюнку 2.4. C і C' — центри обробки даних камери. Точка X — це точка в просторі, яка проєктується для орієнтації x у лівій площині зображення камери та для орієнтації x' у правій площині. Промінь xX є зображенням *переднього плану* точки x . Цей промінь проєктується в площину камери на пряму l' , ця лінія називається епіполярною. Таким чином, можна зробити висновок, що кожна точка x на зображенні лівої камери відповідає епіполярній лінії l' на зображенні правої камери.

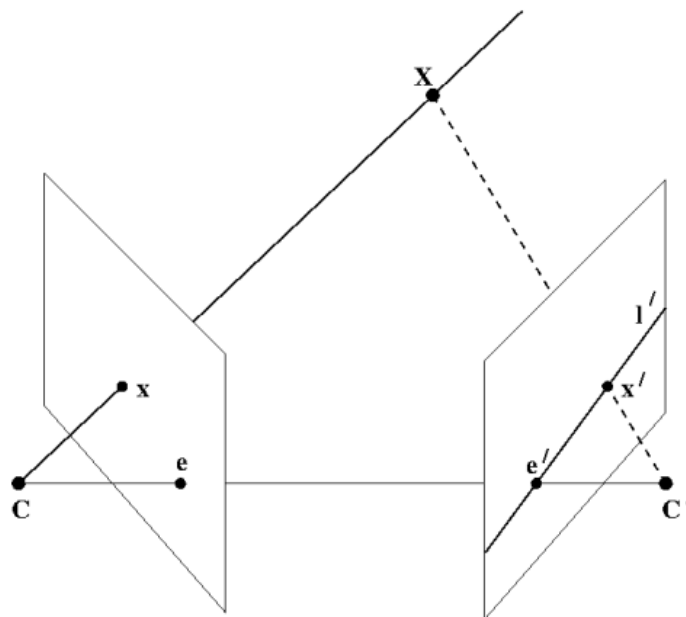


Рис. 2.4. Епіполярна геометрія

Існує матриця F така, що пара точок x і x' є стереопарою тоді і тільки тоді, коли виконується умова:

$$x'^T F x = 0 \quad (2,9)$$

Матриця F залежить тільки від матриць камери. У випадку, коли матриці камер

$$P = K[I|0] \quad (2.10)$$

$$P' = K'[R|t] \quad (2.11)$$

Матрицю F можна розрахувати таким чином:

$$F = K'^{-1T} R K^T [K R^T t] x \quad (2.12)$$

За допомогою цієї матриці розраховуються рівняння епіполярних ліній, тому в точці x лінія матиме таку форму:

$$l' = Fx \quad (2.13)$$

2.3. Характеристика процедури калібрування та обробки зображення

Використання бюджетних камер для вирішення проблем зі стереозором створює кілька додаткових проблем. Це включає функцію калібрування камери (пошук просторових кореляцій між різними зображеннями камери), а також функцію синхронізації кадрових потоків у часі. Завдання калібрування вимагає одноразового вирішення для однієї пари камер, тоді як завдання зіставлення кадрів відбувається безперервно під час роботи камер для

захоплення зображень. Для вирішення проблеми калібрування існує ряд готових алгоритмів, наприклад, Matlab Camera Calibration Toolbox або алгоритми, що входять до бібліотеки OpenCV з відкритим кодом. Калібрування камери виконується шляхом багаторазового фотографування одного і того ж об'єкта (шаблону) обома камерами, на такій фотографії зручно показати набір ключових точок і ліній, за якими відомі фактичні розміри та положення в небі. Для більш точної оцінки часто використовується модель камери-обскури:

$$Z_C \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A * Dist[R, t] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \quad (2.14)$$

де

A - внутрішні параметри калібрування камери, - R матриця обертання 3×3 , T - вектор передачі, $Dist(X)$ - функція спотворення об'єктива, Z_C - додатковий коефіцієнт вирівнювання. Результатом алгоритмів калібрування є внутрішня та зовнішня калібрувальні матриці та вектор значення спотворення лінзи.

Дисторсія - це один із видів оптичних аберацій (дисторсій), який вносить нелінійні спотворення при нормалізації залежно від відстані від лінзи до об'єкта. Загалом спотворення можна виразити так:

$$\vec{R} = b_0 \vec{r} + F_3 r^2 \vec{r}, \quad (2.15)$$

де

r – довжина вектора \vec{r} , F – спотворення заданого порядку. Якщо F має такий самий знак, як і b_0 , спотворення відбувається у вигляді «подушки», в іншому випадку – у вигляді «бочки», як показано на рисунку 2.5.

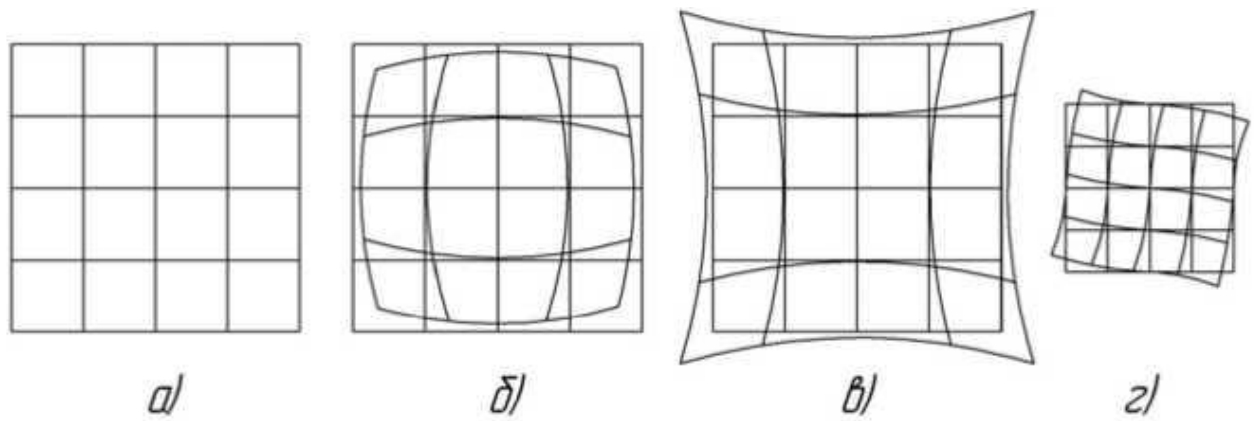


Рис. 2.5. Спотворення форми об'єкта спотворенням: а) тестовий об'єкт (квадратна сітка); б) бочкоподібне спотворення; в) ложеподібне спотворення; г) анізотропні спотворення

Аналітичний або ручний вибір параметрів дозволяє усунути спотворення, але вибране значення буде відповідати лише поточному стану камер. Дані калібрування камери дуже важливі для стереозору, але вони не є вичерпними. Також необхідно визначити співвідношення між системами координат обох камер. Така комбінація буде такою матрицею G , що

$$Pr = G * Pl, \quad (2,16)$$

Пояснити перетворення координат точок Pl в ліву систему камера в Pr правильних координатах.

Отримавши матрицю приводу для кожної з камер, а також матрицю G , можна приступати до обробки. Корекція зазвичай означає створення двох або більше зображень в одній площині [7], зазвичай так, щоб лінії зображення збігалися з епіполярними лініями [8]. Програмна реалізація алгоритму виправлення може бути добре узгоджена для довільної кількості потоків.

Завдання реалізації корекції полягає у встановленні кореляції між епіполярними лініями в стереопарі, для якої буде здійснюватися пошук під час роботи алгоритму стереозгодження.

Визначивши рівняння паралельних прямих стереопари, будемо окрему функцію:

$$E: N^2 \rightarrow N, . \quad (2.17)$$

визначення для кожної пари (Xr, Xl) значення Yr .

Отримана матриця обчислюється один раз і використовується для негайної обробки. Процес підготовки наведено на малюнку 2.6.

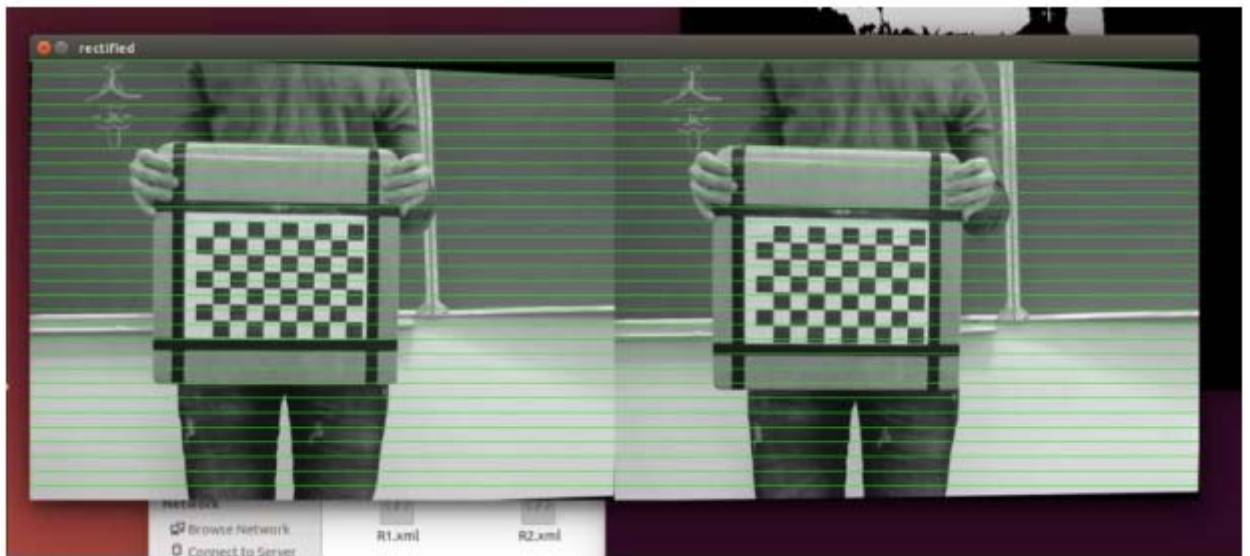


Рис. 2.6. Обробка зображень

Ідея, яка лежить в основі побудови карти глибини на основі стереопар, полягає в тому, що для кожної точки на одному зображенні шукається відповідна точка на іншому. По парі таких точок можна провести триангуляцію і вже в тривимірному просторі визначити координати прототипу. Знаючи ці координати, глибина обчислюється як відстань до площини камери.

Пошук еквівалентних точок відбувається на епіполлярній лінії. Відповідно, щоб полегшити пошук, зображення були вирівняні так, щоб усі епіполярні лінії відповідали сторонам двох зображень (рис. 2.7).

Для кожного пікселя лівого зображення з координатами (x_0, y_0) виконується пошук пікселя правого зображення. При цьому

передбачається, що піксель знаходиться праворуч зображення повинно мати координати, $(x_0 - d, y_0)$ де d - величина, яка називається різницею. Для пошуку відповідного пікселя здійснюється шляхом розрахунку максимального навантаження пам'ять, якою може бути, наприклад, кореляція сусідства пікселів. Результатом є контрастна карта.

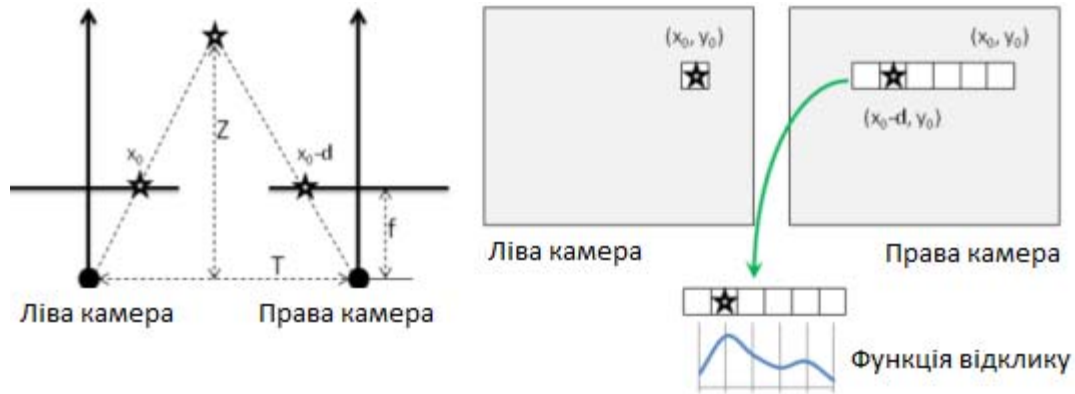


Рис. 2. 7 . Розрахунок карти глибини

Насправді значення глибини обернено пропорційні швидкості зміни пікселів. Використовуючи позначення з лівої половини рисунка 2.7, зв'язок між відстанню та глибиною можна виразити наступним чином:

$$\frac{T-d}{Z-f} = \frac{T}{Z} \rightarrow Z = \frac{fT}{d}, . \quad (2.18)$$

Через зворотну залежність глибини і зміщення роздільна здатність систем стереозйомки, що працюють за цим методом, краще на близьких відстанях і гірше на далеких. Карту глибини можна побачити на малюнку 2.8.



Рис. 2.8. Приклад карти глибини

2.4. Математична модель формування стереопотоку

Вхідними параметрами моделі є набір двох несинхронізованих потоків кадрів (їх зображення, як правило, неепіполярні та некалібровані) і точність синхронізації δ . Вихідним параметром моделі є стереопетля, яка задовольняє вхідні параметри моделі

$$M: X \rightarrow Y, X = (F1, F2, \delta), Y = (Stereo1), . \quad (2.19)$$

2.9 представлена математична модель формування стереопотоку.

Запропонована математична модель формування двокадровий стереопотік від стереокамер заснований на зв'язках з формальними (2.4) - (2.8), (2.19) і використовується при реалізації методів узгодження кадрового потоку, корекції графіки і стереозбігу.

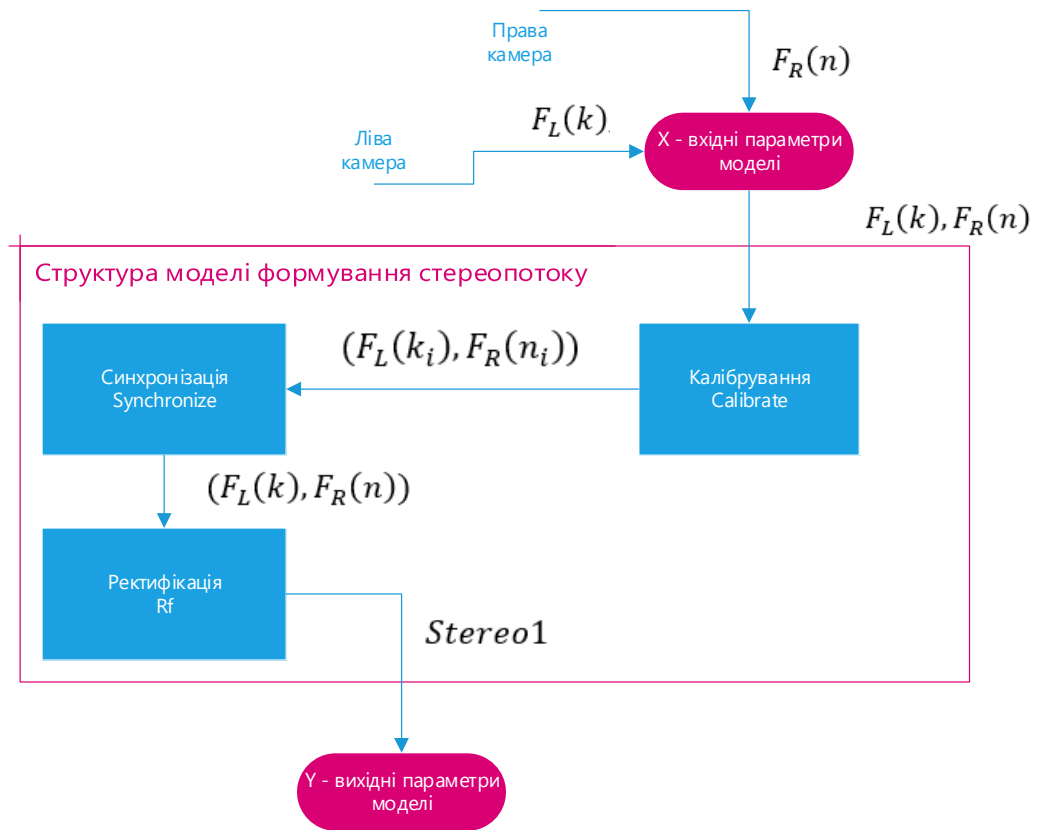


Рис. 2.9. Структурне представлення математичної моделі формування стереопотоку

Висновок другого розділу

1. ідею алгоритму визначення відстані до перешкод мобільного робота на основі обробки стереозображення.
2. Досліджено методи узгодження, калібрування, спотворення та корекції зображень при використанні запропонованого алгоритму визначення відстані до об'єкта;
3. Запропоновано математичну модель формування двокадровий стереопотік зі стереокамер .

РОЗДІЛ

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА ДИЗАЙН МЕТОДУ ВИЯВЛЕННЯ ДЛЯ ВИЯВЛЕННЯ РУХОМИХ КОМПОНЕНТІВ РОБОТА

3.1.1. Впровадження програмного забезпечення

Після створення мобільного робота ми стикаємося з головною проблемою автономності робота - це проходження робота через перешкоди. Через обмежений бюджет вибір методу визначення перешкод обмежився одним способом: за допомогою двох веб-камер. Оскільки нам також потрібні були веб-камери, щоб переглядати те, що бачить робот, ми спершу взяли їх. На малюнку 3.1 зображено робочу схему робота.

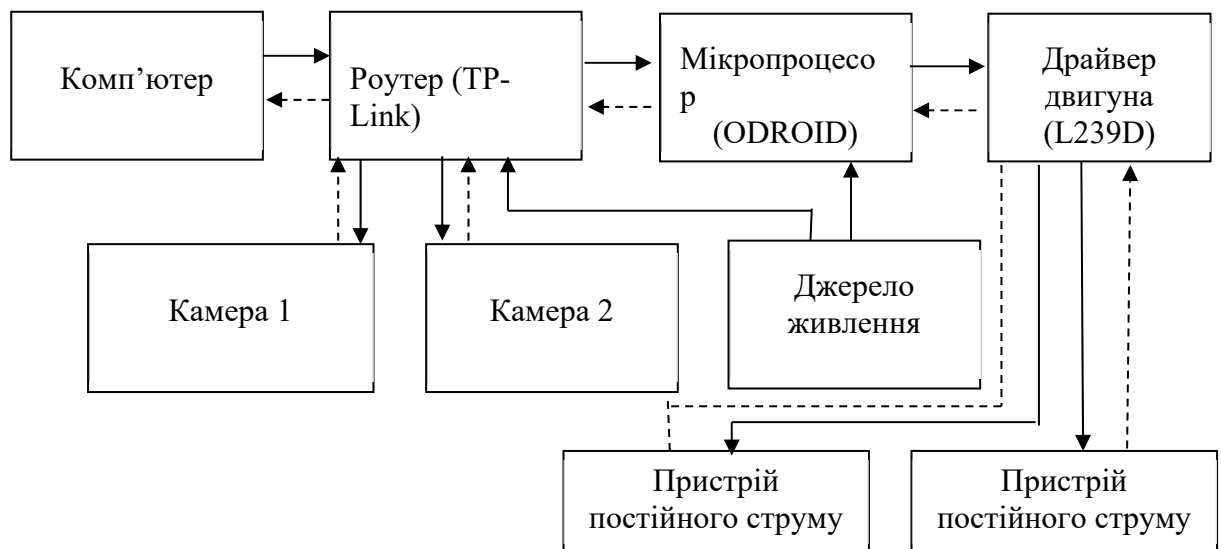


Рис. 3.1.1. Функціональна схема мобільного робота

Камери підключаються до роутера через USB, а роутер передає зображення на комп'ютер через Wi-Fi. Так як роутер, крім передачі відеопотоку на комп'ютер, також отримує від комп'ютера команди для подальшої передачі на мікропроцесор, виникло питання про навантаження на

процесор роутера. Тому також необхідно було знайти оптимальне співвідношення якості відеопотоку та навантаження на процесор. Для цього ми змінили кількість кадрів в секунду (Frame Per Second, FPS). Зі зниженням FPS зменшується навантаження на процесор роутера і збільшується затримка відображення відео на комп'ютері.

Для вузла установки обрано модель ODROID, яка має достатні міцнісні характеристики, найкращі для аналогічного класу обладнання (табл. 3.1). Експериментальна установка представлена на рисунку у другому розділі роботи. В інсталяцію входять: мікрокомп'ютер, дві камери, з'єднані через інтерфейс USB, модуль пам'яті eMMC для зберігання бази даних і системи, блок живлення. При необхідності: модуль wifi для підключення до мережі та монітор для відображення зображення з комп'ютера, підключений через hdmi інтерфейс.

Таблиця 3.1

Порівняльна характеристика одноплатних комп'ютерів

Стандарти (зразок оцінки) .	Raspberry Pi 2	ODROID C1	ODROID U3	ODROI XU3
Дристон-2 цикл	1006,6	1262,8	2300,7	3994,1
Подивіться двічі	1076.1	1318,8	2217,9	3074,7
Валун (x3) .				
Nbench 2.2.3 Integer (x40) .	840.0.0	1208,0	1648,0	2451,1
Стенд 2.2.3 FP (x100) .	809.0.0	1050,0	1860,0	2536,9
SysBench (1/T*50000).	669.3	902.2	1497,0	2891,8
пам'яті (mbw 100:MiB/s) .	557.1	1069,6	736,6	3243,8

Програма впровадження запропонованого алгоритму розроблена з урахуванням наступних вимог:

- кількість знімків зображення під час калібрування - не більше 1 (однієї) стереопари в секунду;
- формат вхідного зображення - *.jpeg / *.pgm
- розмір не більше 640*480;
- обробка камери – сумісність;
- операційна система сімейства Unix;
- пристрій - Odroid XU3.

Технічні характеристики Odroid XU3, взяті з офіційного сайту виробника:

- ЦП - Exynos 5422 Large Octa.SMALL HAND CortexA15 @ 2.0 GHz;
- GPU - Mali-T628 MP6 (OpenGL ES 3.0 / 2.0 / 1.1 і OpenCL 1.1 Full-Size);
- Оперативна пам'ять - 2 ГБ LPDDR3 RAM на 933 МГц (пропускна здатність пам'яті 14,9 ГБ/с).

Під час тестування використовувалися два набори зображень, отриманих з камер різної якості. У першому випадку використовувалася веб-камера Logitech HD Webcam C270, у другому – камери Cognex, що спеціалізуються на комп'ютерному зрінні та обробці зображень, відомі високоякісними можливостями фотографування та обробки зображень прямо в камері завдяки вбудованому процесору. .

Бібліотека OpenCV потрібна для повноцінної роботи проекту в цілому та окремих його компонентів. OpenCV випущено за ліцензією BSD і є безкоштовним як для академічного, так і для комерційного використання. Ця бібліотека містить алгоритми обробки зображень, математичні алгоритми загального призначення з відкритим кодом.

Він має інтерфейси для C++, C, Python і Java і підтримує ОС Windows, Linux, Mac OS, Android і IOS. OpenCV розроблено для підвищення ефективності обчислень і зосереджено на програмах реального часу. Бібліотека, написана на оптимізованому C/C++, може скористатися

перевагами багатоядерної обробки. Разом із OpenCL він може скористатися перевагами апаратного прискорення основної гетерогенної обчислювальної платформи. OpenCV, прийнятий у всьому світі, нараховує понад 47 тисяч людей із спільноти користувачів із приблизною кількістю завантажень понад 9 мільйонів людей. Застосування варіюються від інтерактивного мистецтва до роботизованих розширень.

Бібліотека OpenCV містить велику кількість модулів для різних функцій. У його роботі крім базового модуля `opencv_core` використовувався:

- `Opencv_highgui` – інтерфейс імпорту-експорту зображень.
- `Opencv_objdetect` - виявлення об'єктів на зображенні.
- `Opencv_calib3d` - калібрування камери, пошук стереозв'язку, тривимірна обробка даних.
- `Opencv_video` - аналіз руху та відстеження об'єктів у відеопотоці.

Усі програмні модулі для цього проекту написані мовою C++ із використанням компілятора `g++` для операційної системи Linux. Інструмент `Make` використовується для компіляції та запуску кожного з модулів. Ця утиліта за допомогою команд, визначених у файлі `makefile`, допомагає скомпілювати зазначені в цьому файлі бібліотеки, необхідні для роботи програми, а також скомпілювати та скомпілювати файл запуску.

Make-файл будується за шаблоном:

Призначення: залежність;

команди [Tab];

Простий приклад можна подати так:

все:

`g++` для калібрування.cpp -о калібрувати

де `g++` — компілятор, який використовується, `calibration.cpp` — скомпільований файл, що містить програмний код, `calib` — ім'я вихідного файлу, створеного під час компіляції.

Крім того, у цьому файлі можна вказати залежності від інших файлів, якщо проект має декілька файлів. Можна вказати прапорці колектора.

Make-файл для цього проекту та інтерфейс для всіх бібліотек для роботи з зображеннями, потокового відео, побудови карт глибини та подальшого визначення відстані до об'єкта виглядає так:

```
CC = g++
CFLAGS=-c -gCFLAGS+=-Fence -Wno-deprecatedLDFLAGS =PROPERTIES =
main.cppOBJECTS = $(PROPERTIES:.cpp=.o) DESIGNED = qwe.outCFLAGS+= `pkg-config
--cflags pcl_common-1.7 pcl_registration -1.7 pcl_transparency-1.7
opencv`CFLAGS+= `pkg-config --cflags eigen3
CFLAGS+= -I/usr/include/vtk-5.8
CFLAGS+= -I/usr/include/eigen3#CFLAGS += -I/usr/include/pcl-1.7LDFLAGS +=
`pkg-config --libs pcl_common-1.7 pcl_registration -1.7 pcl_visualization-1.7
opencv`LDFLAGS+= -lboost_system -lboost_threadLDFLAGS+= -lusb-1.0 -lpthread -
lz -lvtkCommon -lvtkGraphics -lvtksys -lvtkCharts -lvtkDICOMParserLDFLAGS+= -
lvtkFiltering -lvtkGeovis -lvtkHybrid -lvtkIO -lvtkImaging -lvtkInfovis -
lvtkParallel -lvtkRenderingFLAGS + = -L/lib64all: $(джерела)
$(CREATED)$(CREATED): $(CONTENTS)$(CC) $(CONTENTS) -o $$@ $(LDFLAGS)%.o: %.cpp$
(CC) $(CFLAGS) $< -o $$@clean:rm $(STUFF) $(ГОТОВО)
```

Ця робота складається з чотирьох основних модулів. Модуль для паралельного захоплення двох відеопотоків з веб-камер, модуль, який виконує стереокалібрування камер, модуль, який одночасно виконує обробку отриманого зображення та побудову карти глибини, а також модуль визначення відстані для виходу на об'єкт.

Модуль захоплення відео.

Ця програма обробляє відеопотоки:

```
CvCapture* capLeft = cvCreateCameraImage(0);
assert( capLeft );CvCapture* capRight = cvCreateCameraImage(1);
assert( capRight ) .
```

А потім виконує одночасне збереження зображень з двох камер:

```
while ( true )
{ time_t cur_time=time (NULL)+2; char [
200 ] = {};
sprintf(buff, "filenameL%d.jpg" ,i);
char buff1[200] = {};
sprintf(buff1, "filenameR%d.jpg" ,i);
cvSaveImage(buff, frameLeft);cvSaveImage(buff1, frameRight);i++;
if (cur_time >= stop_time) break ;
} .} .
```

Модуль калібрування стерео. Цей модуль приймає зображення як вхідні дані шахової дошки з двома камерами, виконує калібрування шляхом знаходження кутів, знаючи їх фізичні розміри, які можна побачити на малюнку 3.2.



Рис. 3.2. Щоб отримати кути

При порівнянні систем координат двох камер розраховується базисна матриця, яка буде використана для побудови карти глибини для цих зображень. Також тут відбувається вирівнювання зображення з епіполлярними лініями і усунення дисторсії (дисторсії). Пошук кутів на тестовій дошці виконується так:

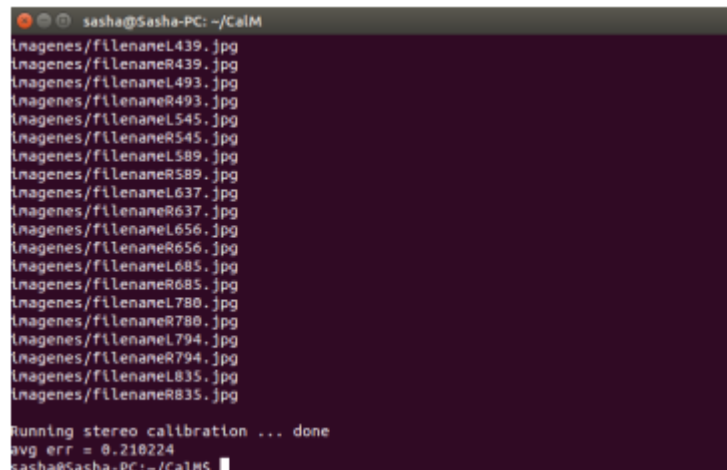
```
//Отримуємо кути шахівниці:
for ( int s = 1; s <= maxScale; s++ )
{IplImage* timg = img;
if ( s > 1 )
{timg = cvCreateImage(cvSize(img->width*s, img->height*s),
img->depth, img->nChannels );cvResize( img, timg, CV_INTER_CUBIC ); }
результат = cvFindChesboardCorners( timg, cvSize(nx, ny), &temp[0], &count,
CV_CALIB_CB_ADAPTIVE_THRESH |CV_CALIB_CB_NORMALIZE_IMAGE);
if ( timg != img )
cvReleaseImage( &timg );
if (результат || s == maxScale )
for ( j = 0; j < count; j++ )
{temp[j].x /= s;temp[j].y /= s;}
if (результат)
break ;
} .} .
```

Одним з основних модулів програми, що розробляється, є модуль калібрування стереокамери. Оскільки цей алгоритм, як уже згадувалося раніше, заснований на виявленні заздалегідь визначеної закономірності, а саме аналізі в шаховому порядку з метою знаходження всіх кутів, заданих обома

камерами, можна обчислити похибку значень, отриманих з отримані матриці аддукції, заздалегідь знаючи фізичні обмеження використовуваної структури.

Ми можемо оцінити середню похибку обчислень і продуктивність функції StereoCalibrate порівняно з фактичними фізичними межами шахової дошки. Це значення вимірюється в пікселях і вказує на похибку між фактичним положенням точки на док-панелі та положенням, обчисленим за допомогою документованого алгоритму.

Функція виявлення помилок повертає середнє арифметичне всіх калібрувальних зображень. Це значення повинно бути якомога ближче до нуля, значення похибки від 0 до 0,49 вважається нормальним, отримане в результаті операції значення склало 0,210224, що вважається більш допустимим. Результати розрахунків представлені на рисунку 3.3.



```
sasha@sasha-PC: ~/CalM
lnagenes/filenaneL439.jpg
lnagenes/filenaneR439.jpg
lnagenes/filenaneL493.jpg
lnagenes/filenaneR493.jpg
lnagenes/filenaneL545.jpg
lnagenes/filenaneR545.jpg
lnagenes/filenaneL589.jpg
lnagenes/filenaneR589.jpg
lnagenes/filenaneL637.jpg
lnagenes/filenaneR637.jpg
lnagenes/filenaneL656.jpg
lnagenes/filenaneR656.jpg
lnagenes/filenaneL685.jpg
lnagenes/filenaneR685.jpg
lnagenes/filenaneL780.jpg
lnagenes/filenaneR780.jpg
lnagenes/filenaneL794.jpg
lnagenes/filenaneR794.jpg
lnagenes/filenaneL835.jpg
lnagenes/filenaneR835.jpg
Running stereo calibration ... done
avg err = 0.210224
sasha@sasha-PC:~/CalM$
```

Рис. 3.3. Сума помилки

Модуль побудови карти глибини. Щоб побудувати карту глибини, необхідно завантажити всі матриці, отримані на попередньому етапі калібрування камери, і розміри вхідних зображень. Далі ви можете дослідити стереосумісність.

```
//Отримати карту відмінностей
if ( !isVerticalStereo )
pair = cvCreateMat( imageSize.height,imageSize.width * 2,CV_8UC3 );
else
pair = cvCreateMat(imageSize.height * 2
,imageSize.width,CV_8UC3);
```

```

//Налаштування для отримання стереосинхронізації
CvStereoBMState * BMState =
cvCreateStereoBMState(); assert (BMState!= 0); BMState -> preFilterSize = 41;
BMState -> preFilterCap = 31; BMState -> SADWindowSize = 41; BMState ->
MinDisparity =- 64; BMState -> numberOfDisparities = 128; =10;BMState->
SelectCount=15;
for ( i = 0 ; i < nframes ; i++ )
{ IplImage *
img1=cvLoadImage(imageNames[0][i].c_str(),0);IplImage*img2=cvLoadImage(imageNa
mes[1][i]. ]. c_str( ),0);
if ( img1 && img2 )
{ printf( " done\n" );
Частина CvMat;cvRemap( img1, img1r, mx1, my1);cvRemap( img2, img2r, mx2, my2);
if ( !isVerticalStereo ||
useUncalibrated != 0 ){cvGetStereoCorrespondenceBM(img1r, img2r,
disp,BMState);cvNormalize( disp, vdisp, 0, 256,CV_MINMAX );cvNamedWindow(
"contrast" );
cvShowImage( "контрастність" , vdisp); char [
200 ] = {};
sprintf(buff, "filename%d.pgm" ,i);
cvSaveImage(buff,vdisp );}
if ( !isVerticalStereo )
{ cvGetCols( pair, &component, 0, 640);cvCvtColor( img1r,
&component,CV_GRAY2BGR );cvGetCols( пара, &component, 640,640*2
);cvCvtColor(img2r). , &part ,CV_GRAY2BGR );,CV_GRAY2BGR );
для ( j = 0; j < 480; j += 16 )
cvLine( пара, cvPoint(0,j),cvPoint(640*2,j),CV_RGB(0,255,0));}cvShowImage(
"виправлено" , обидва);
if ( cvWaitKey ( ) == 27)
перерва ;
}cvReleaseImage( &img1);cvReleaseImage( &img2);}

```

Як видно з наведеного коду, існує певна форма BMState, яка зберігає кількість параметрів, необхідних для отримання стереокореляції. Для зручності написана окрема програма з користувацьким інтерфейсом, що дозволяє змінювати ці параметри вручну, спостерігаючи, як змінюється отримана карта глибини в реальному часі. Ці параметри, встановлені вручну за допомогою повзунків, потім можна зберегти та передати програмі збірки, яка використовуватиме їх під час автоматичного створення карти. Дані стосуються поточного стану камер. Карта глибини, отримана після калібрування за цим алгоритмом, представлена на рисунку 3.4.

Звичайно, якість карти глибини безпосередньо залежить від якості камери, що використовуються для зйомок. Чим вище їх множення і чим більше схожі матриці обох камер під час стереозйомки, тим вища якість карти глибини.



Рис. 3.4. Карта глибини

Модуль визначення відстані до об'єкта. Розрахунок відстані до спостережуваного об'єкта здійснюється за допомогою інформації про слабкість його структури. Вони були в цьому створено умови, за яких на рис об'єкт розташований в тій же формі щоб запобігти перешкодам від несприятливих циклів. Розраховується середня інтенсивність кольору всіх контурних точок об'єкта на карті глибини, після чого отримане значення множиться на розрахований коефіцієнт для приведення в метровий масштаб, після чого результат виводиться на консоль.

```
статичний простір inDistance ( int, void* ) .
{
static const ScaleColors [ ] =
{
Масштаб (0,0,0), .
255.0.0 ), .
255 128,0 ), .
255 255,0 ), .
0,255,0 ), .
0,128,255 ), .
0,255,255 ), .
0,0,255 ), .
255.0.255 )
};
int maskSize = voronoiType >= 0? DIST_MASK_5 : maskSize0;
int distType = voronoiType >= 0? DIST_L2 : Введіть dist0;
Матовий край = сірий >= edgeThresh, dist, beard, dist8u;
if( voronoiType < 0 ) .
```

```

distanceTransform (напрямок, відстань, distType, maskSize);
деякі
distanceTransform ( edge , dist , labels , distType , maskSize , voronoiType
);
if( voronoiType < 0 ) .
{
// починаємо "малювати" результат перетворення відстані
регіон *= 5000;
row (плоский, 0,5, круглий);
Mat dist32s, cross8u1, cross8u2;
відст. convertTo (dist32s, CV_32S , 1, 0,5);
dist32s &= Scope::all(255);
dist32s. convertTo (dist8u1, CV_8U , 1, 0);
dist32s *= -1;
dist32s += Scope::all(255);
dist32s. convertTo (dist8u2, CV_8U );
imshow ("Віддалені карти", dist8u);
} .} .
} .} .
повернути 0;
} .} .

```

3.2. Експериментальні дослідження

У цій частині роботи ми пройдемося по етапах алгоритму визначення відстані до перешкод і розглянемо артефакти, отримані з кожного кроку описаного вище алгоритму. Також побудуємо графік залежності похибки визначення контрасту від відстані до об'єкта спостереження.

Наведемо візуальні зображення результатів роботи алгоритму та побудуємо графік залежності похибки визначення контрасту від відстані до об'єкта спостереження.

Під похибкою визначення контрасту ми маємо на увазі модуль різниці значення контрасту, який відповідає розрахованій відстані (на основі мінімальної відстані, фокусної відстані та значення контрасту), і значення різниці k_u , що відповідає фізичній відстані .

Таблиця 3.2

Практичні аспекти від дизайну стереопари до спостережень

Об'єкт	Різниця від стереопари до об'єкта (м) .
Поверхня столу	2.3
Перша книга	3.11
Друга книга	3.45
Книга третя	3.72
дошка	4.15

Таблиця 3.3

Кількість відмінностей точок спостережень, витягнутих з карти різниць

Об'єкт	Координата точки об'єкта (x, y) .	Важливість відмінності
Поверхня столу	(324, 274)	56.5
Перша книга	(508, 207)	42.0.0
Друга книга	(415, 199)	37,6875
Книга третя	(335, 200)	35,0625
дошка	(230, 116)	31.0.0

На рисунку 3.5 показано зображення з правої камери (етап збору кадрів з камери та зображення шахової дошки).

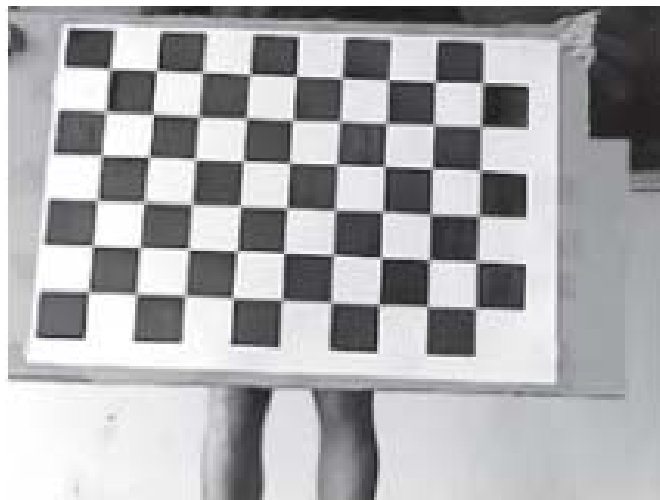


Рис. 3.5. Зображення з правої камери

На малюнку 3.6 показано зображення з лівої камери (етап збору кадрів з камери на зображенні шахової дошки).

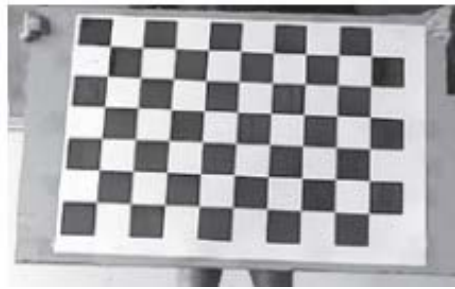


Рис. 3.6. Зображення з лівої камери

3.7 показано зображення з лівої камери та отримане зображення (етап пошуку вузлів зображення шахової дошки на зображеннях).

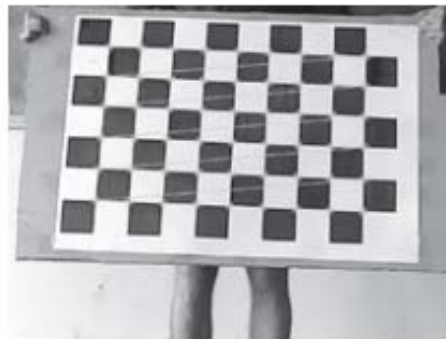


Рис. 3.7. Зображення з лівої камери та отримане зображення (крок пошуку вузлів зображення шахової дошки на зображеннях)

На малюнку 3.8 показано оброблене зображення з правої камери зі знятим спотворенням (Умова обробки та усунення спотворень).

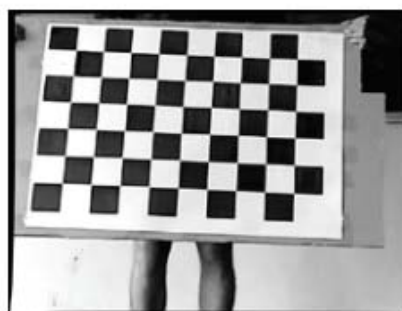


Рис. 3.8. Виправлене зображення з правої камери з усуненням спотворень
(Етап корекції та усунення спотворень)

На малюнку 3.9 показано оброблене зображення з лівої камери зі знятим спотворенням (Стан обробки та видалення спотворень).

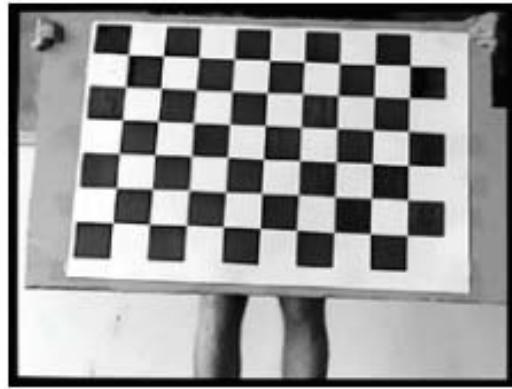


Рис. 3.9. Оброблене зображення з лівої камери з вилученим спотворенням
(статус обробки та видалення спотворень)

На малюнку 3.10 зображено схему сцени з об'єктами спостереження (стіл, три книги та дошка, розташовані на відстані один від одного).



Рис. 3.10. Ескіз сцени з об'єктами спостереження (стіл, три книги і
дошка, розташовані на відстані один від одного).

На малюнку 3.11 показано карту контрастності зображення з правої та лівої камер (стіл, три книги та дошка, розташовані далеко одна від одної).



Рис. 3.11. Контрастна карта зображень з правої та лівої камер (стіл, три книги та дошка, розташовані далеко один від одного).

3.12 показано залежність похибки визначення контрасту від відстані до об'єкта спостереження.



Рис. 3.12. Залежність похибки визначення контрасту від відстані до об'єкта спостереження.

З графіка (рисунок 3.12) видно, що максимальне значення похибки визначення різниці відстані до спостережуваного об'єкта від 2,30 до 4,15 метрів становить лише 1,2, що вказує на хорошу якість різної карти ку.

Розроблений алгоритм розрахунку відстані до спостережуваного об'єкта за зображеннями зі стереопари показує хорошу ефективність, а наведені розрахунки похибки визначення різниці показують залежність від відстані до спостережуваного об'єкта. . . .

Підсумок третьої глави

В даному розділі проводяться практичні застосування розроблених методів і алгоритмів з використанням програмних модулів для реалізації алгоритму визначення відстані до перешкод для мобільного робота на основі обробки стереозображення. Здійснено впровадження програмного забезпечення, проведено затвердження та тестування.

На основі аналізу та технічної оцінки результатів впровадження програмних модулів обґрунтовано успішне впровадження розробленого алгоритму в системи підтримки прийняття рішень.

НАДЗВИЧАЙНІ ПРОДУКТИ

Таким чином, під час виконання магістерського дослідження всі поставлені завдання були вирішені. Розроблено багатовимірну систему картографування глибини, яка складається з модуля для одночасної зйомки зображень з двох камер, модуля калібрування, модуля побудови карти та модуля визначення відстані до об'єкта. . . . Також був написаний модуль для обробки отриманих карт глибини в режимі реального часу шляхом ручного вибору параметрів.

У цій роботі розроблено алгоритм розрахунку відстані до спостережуваного об'єкта на основі зображень зі стереопари. Наведені розрахунки похибки визначення дисперсії в залежності від відстані до об'єкта спостереження.

Написані алгоритми оцінювали за допомогою моделювання синтетично, а потім у реальному часі з використанням даних, отриманих від інтегрованих камер.

Розглянуто методи виявлення перешкоди, обчислення її меж, відносної швидкості та кутового положення. Розглянуто задачу визначення оптичного потоку та наведено порівняльні результати різних методик.

Визначено оптимальні умови для роботи даної системи, обрано розташування камер одна відносно одної для досягнення найкращих результатів, а також апаратну складову обробки даних.

На основі аналізу та технічної оцінки результатів впровадження програмних модулів обґрунтовано успішне впровадження розробленого алгоритму в системи підтримки прийняття рішень.

АНАЛІЗ ICORIS НА ДЕШЕВІ ПРОДУКТИ

1. Грицик В. В. Методи проектування систем нейрокомп'ютерного обладнання для мобільних робототехнічних систем / Грицик В. В., Цмоць І. Г., 156 Теслюк В. М. // Доповіді НАН України. – Київ, 2013. – Число 1. – С. 30-36.
2. Самойлов, В. Д. Побудова комп'ютерної моделі / В. Д. Самойлов. – К.: Наукові перспективи. - 2007. - 198 с.
3. Субботін С. О. Представлення та обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник / Субботін С. О. – Запоріжжя: ЗНТУ, 2008. – 341 с.
4. Грем І. Об'єктно-орієнтовані методи. Принципи і практика / Ян Грем. – Вільямс, 2004. – 880 с. 75. Міркес Е. М. Нейрологія: Підручник. посібник для студентів / Е. М. Міркес. – Красноярськ: ІВЦ КГТУ, 2002. – 347 с.
5. Бідюк П. І. Комп'ютеризовані системи підтримки прийняття рішень / П. І. Бідюк, О. П. Гожій, Л. О. Коршевніюк. – Київ, 2012. – 379 с.
6. Лесна Н. С. Інтелектуальний аналіз даних / Н. С. Лесна, В. Б. Репка, Т. Б. Шатовська. – Харків: Харківський національний електронний радіоун-т, 2003. – 110 с.
7. Ємельянов В. В. Теорія і практика еволюційного моделювання / Ємельянов В. В., Курейчик В. В., Курейчик В. М. – М.: ФІЗМАТЛИТ, 2003. – 432 с.
8. Новотарський М. А. Штучні нейронні мережі: обчислення / М. А. Новотарський, Б. Б. Нестеренко // Збірник наукових праць Інституту математики НАН України. - Вип. 50. – К. : Інститут статистики НАН України, 2004. – 408 с.