

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки  
Кафедра автоматизації та управління в технічних системах**

«На правах рукопису»  
УДК 681.03

«До захисту допущено»

В. о. завідувача кафедри

\_\_\_\_\_ О.І. Ролік

«\_\_\_» \_\_\_\_\_ 2018 р.

## **Магістерська дисертація**

**на здобуття ступеня магістра**

**зі спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології  
на тему: «Засоби розпізнавання об'єктів в системі комп'ютерного зору для цифро-  
вої нарізки матеріалів»**

Виконав:

студент II курсу, групи ІА-61м  
Туманов Владислав Валерійович \_\_\_\_\_

Керівник:

Проф., д. ф.-м. н., проф.,  
Дорошенко А. Ю. \_\_\_\_\_

Рецензент:

Доц., к. ф.-м. н., доц.,  
Ігнатенко О. П. \_\_\_\_\_

Засвідчую, що у цій магістерській дисертації не-  
має запозичень з праць інших авторів без відпові-  
дних посилань.

Студент \_\_\_\_\_

Київ – 2018 року

## РЕФЕРАТ

Дипломна робота освітньо-кваліфікаційного рівня “магістр” на тему “Засоби розпізнавання об’єктів у системі комп’ютерного зору для цифрової нарізки матеріалів”: 107 с., 24 рис., 26 табл., 2 додатки, 16 джерел.

Об’єкт дослідження – система комп’ютерного зору для цифрової нарізки.

Предмет дослідження – засоби комп’ютерного зору, які можна застосувати для розпізнавання та позиціонування об’єктів на поверхні станка для цифрової нарізки.

Мета роботи – аналіз можливостей сучасних засобів комп’ютерного зору з метою їх максимально ефективного застосування у системі комп’ютерного зору для сфери цифрової нарізки матеріалів.

Системи комп’ютерного зору, що спеціалізуються на розпізнаванні, класифікації та ідентифікації об’єктів, постійно знаходять нові сфери для застосування та є одним з найбільш досліджуваних напрямків у розвитку комп’ютерної техніки. Проте у сфері цифрової нарізки матеріалу комп’ютерний зір усе ще застосовується досить обмежено.

У даній дисертації для сфери серійної цифрової нарізки промаркованих матеріалів розроблена система комп’ютерного зору, яка використовує найсучасніші засоби розпізнавання та позиціонування об’єктів з метою максимально автоматизувати процес нарізки та мінімізувати людське втручання у нього.

Для дослідження та побудови програмної частини системи використовувались засоби відкритої бібліотеки комп’ютерного зору OpenCV та інші інструменти розробки з відкритим кодом.

Найбільш вірогідним прогнозом подальшого розвитку розробленої технології є її вихід на світовий ринок та впровадження її в якості модулів майбутніх станків для цифрової нарізки.

СИСТЕМИ КОМП’ЮТЕРНОГО ЗОРУ, ПОЗИЦІОНУВАННЯ ОБ’ЄКТІВ, ІДЕНТИФІКАЦІЯ ОБ’ЄКТІВ, КАЛІБРУВАННЯ КАМЕРИ, ПОШУК ЗА ЗРАЗКОМ, УЗГОДЖЕННЯ МНОЖИН ТОЧОК.

## ABSTRACT

Diploma work of educational qualification level "Master" on the topic "Objects recognition tools in computer vision system for digital cutting of materials": 100 p., 21 figures, 22 tables, 1 supplement, 14 sources.

The object of research – computer vision system for digital cutting.

The subject of research – the means of computer vision, which can be used to recognize and position objects on the surface of the machine for digital cutting.

The purpose of the work is to analyze the possibilities of modern means of computer vision in order to maximally efficiently use them in the computer vision system for the digital material cutting sector.

Systems of computer vision specializing in recognition, classification and identification of objects constantly find new areas for application and are one of the most studied areas in the development of computer technology. However, in the field of digital cutting of material, computer vision is still used very limited.

In this dissertation, for the field of serial digital cutting of labeled materials, a computer vision system is developed that uses the most advanced means of recognition and positioning of objects in order to maximize the automation of the cutting process, and minimize human interference with it.

To explore and build the program part of the system, the OpenCV open computer vision library and other open source tools were used.

The most probable forecast for the further development of the designed technology is its output on the world market and its implementation as modules for future digital cutting machines.

COMPUTER VISION SYSTEMS, OBJECTS POSITIONING, OBJECTS IDENTIFICATION, CAMERA CALIBRATION, SEARCHING BY TEMPLATE, POINT SET REGISTRATION

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	7
ВСТУП.....	8
1 ОГЛЯД СУЧАСНОГО СТАНУ СФЕРИ ЦИФРОВОЇ НАРІЗКИ.....	10
1.1 Основні проблеми в сфері цифрової нарізки та методи їх вирішення.....	10
1.2 Застосування комп'ютерного зору в сфері цифрової нарізки.....	17
2 ЗАДАЧІ, МОДУЛІ ТА ІНСТРУМЕНТИ РОЗРОБЛЮВАНОЇ СИСТЕМИ .....	19
2.1 Постановка задачі та вхідні дані.....	19
2.2 Основні модулі розроблюваної системи.....	20
2.3 Технічна база та інструменти розробки системи комп'ютерного зору.....	22
2.3.1 Необхідна технічна база розроблюваної системи .....	22
2.3.2 Обґрунтування вибору C++ в якості мови програмування для розробки ядра системи .....	23
2.3.3 Використання OpenCV як джерела потужних засобів комп'ютерного зору .....	26
2.3.4 Можливості використання апаратного прискорення та паралельних обчислень з алгоритмами OpenCV .....	29
2.3.5 Можливості Qt-фреймворку для побудови кроссплатформового ядра та графічного інтерфейсу.....	33
2.3.6 Microsoft Visual Studio 2015 як середовище розробки ядра та демонстраційної версії програми .....	35
3 КАЛІБРУВАННЯ І ПОЗИЦІОНУВАННЯ ОБ'ЄКТІВ НАРІЗКИ .....	38

3.1 Оптичні дисторсії та їх вплив на зображення різальної поверхні плоттера ..	38
3.2 Калібрування камери для усунення оптичних спотворень на фотографіях ..	44
3.3 Створення системи позиціонування на основі калібрувальної сітки .....	47
3.4 Реалізація процедури калібрування системи комп'ютерного зору .....	51
4 ОБРОБКА ЗОБРАЖЕННЯ ТА ПОШУК РЕЄСТРАЦІЙНИХ МАРОК .....	55
4.1 Засоби OpenCV для пошуку реєстраційних марок.....	55
5 ІДЕНТИФІКАЦІЯ ОБ'ЄКТІВ НАРІЗКИ.....	64
5.1 Алгоритми узгодження множин точок .....	64
5.2 Процедура ідентифікації об'єктів цифрової нарізки.....	67
6 ДЕМОНСТРАЦІЙНА ПРОГРАМА .....	80
7 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ .....	86
7.1 Опис ідеї проекту .....	86
7.2 Технологічний аудит ідеї проекту.....	89
7.3 Аналіз ринкових можливостей запуску стартап-проекту.....	90
7.4 Розроблення ринкової стратегії проекту .....	98
7.5 Розроблення маркетингової програми стартап-проекту.....	101
ВИСНОВКИ.....	105
РЕКОМЕНДАЦІЇ.....	106
ПЕРЕЛІК ПОСИЛАНЬ .....	107

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

СКЗ – система комп'ютерного зору

ЦНМ – цифрова нарізка матеріалів

РМ – реєстраційна марка

ПЗ – програмне забезпечення

API (англ. Application program interface) – прикладний програмний інтерфейс

CPD (англ. Coherent Point Drift) – узгоджений дрейф точок

Плоттер (англ. Plotter) – у контексті даного дослідження – станок для цифрової нарізки матеріалів

*px* – піксель

## ВСТУП

Розвиток комп'ютерної техніки в останні роки зумовив значне посилення ролі систем комп'ютерного зору в різних галузях промисловості. Задачі розпізнавання та класифікації об'єктів, які раніше могла розв'язати лише людина, нині успішно розв'язуються за допомогою обчислювальної техніки. Не обійшов технічний прогрес і галузь цифрової нарізки матеріалів (ЦНМ). Вмонтовані в різець камери вже давно служать для пошуку контуру об'єкта нарізки та складання правильної траєкторії руху різця.

Проте сучасний стан розвитку засобів комп'ютерного зору дозволяє замислитись над розробкою такого методу розпізнавання та позиціонування об'єктів на різальній поверхні станка, який би дозволив за допомогою зображення всієї поверхні швидко розпізнати та визначити положення розміщених на ній промаркованих матеріалів для нарізки, без застосування вмонтованої в різальний механізм камери.

Метою даної дисертації є розробка системи комп'ютерного зору, яка, використовуючи найсучасніші технології та алгоритми розпізнавання, дозволить з високою точністю, за допомогою лише одного знімку робочої поверхні станка для ЦНМ, визначити на основі заздалегідь заданих шаблонів тип та положення відносно точки відліку різця, розташованого на ній, та промаркованих відповідно до шаблону об'єктів нарізки.

У ролі об'єкта дослідження в даній дисертації виступає розроблювана система комп'ютерного зору для ЦНМ. Предметом дослідження є засоби комп'ютерного зору, використання яких дозволить максимально прискорити та автоматизувати процеси розпізнавання та позиціонування об'єктів для цифрової нарізки.

У процесі розробки системи наскрізним методом дослідження виступає експеримент, оскільки саме завдяки йому можна було визначити, які засоби комп'ютерного зору є більш ефективними в тій чи іншій ситуації. За відсутності доступу до реальних станків ЦНМ, у якості матеріалів для експериментів часто виступали моделі робочих поверхонь станків з розміщеними на них об'єктами нарізки. Для порівняння результатів експериментів проводились вимірювання таких параметрів, як швидкодія, об'єм використовуваної пам'яті та розмірність вхідних даних, що визначає складність задачі розпізнавання.

Наукова новизна даного дослідження полягає в принципово новому підході до розпізнавання та позиціонування об'єктів для нарізки. Раніше для цього використовувалась локальна камера, закріплена на різці станка, - різальному механізму необхідно було пройти вздовж усього контуру об'єкта нарізки, щоб скласти повну і точну картину для подальшого нарізання, і тільки після цього почати безпосередньо нарізку. Таким чином виконувалась подвійна робота механізму, що призводило до його вдвічі швидшого зношування. Новий підхід потребує лише виконання одного знімку всієї різальної поверхні глобальною камерою, розташованою над нею. Використовуючи лише цей знімок, розроблена програма визначає всі сфотографовані об'єкти нарізки та їх положення в системі координат різця. Такий підхід економить як ресурс механізму різця, так і дорогоцінний час, що є визначними показниками у промислових масштабах.

Практичне застосування розробленої системи можливе лише для промаркованих спеціальними реєстраційними марками об'єктів нарізки, для яких можна застосувати шаблон. Утім, маркування об'єктів є досить поширеною практикою в серійній нарізці текстилю, паперу та інших матеріалів у промислових масштабах.

Загальна концепція системи представлена у вигляді статті в журналі «Проблеми програмування», а принцип роботи модуля позиціонування та калібрування подано у формі тези на конференції «Winter InfoCom Advanced Solutions 2017».



## 1 ОГЛЯД СУЧАСНОГО СТАНУ СФЕРИ ЦИФРОВОЇ НАРІЗКИ

### 1.1 Основні проблеми в сфері цифрової нарізки та методи їх вирішення

Переваги ЦНМ у максимально продуктивному безперервному процесі друку-нарізки не підлягають сумніву. Тому нові моделі цифрових різальних систем (плоттерів) та їх виробники продовжують виходити на світовий ринок. Галузі промисловості, що напряму пов'язані з друком на різноманітних матеріалах, були і залишаються головними користувачами та споживачами на ринку плоттерів. Для них свою найкращу продукцію готові запропонувати такі фірми як Zünd, EskoArtwork Kongsberg, Summa, Aristo та інші відомі в описуваній сфері бренди.

Більшість сучасних плоттерів – це масивний стіл з робочою поверхнею, на якій розміщуються об'єкти для їх подальшої нарізки. Поперек столу над робочою поверхнею нависає перекладина, на якій закріплений різучий механізм разом з камерою. Перекладина може вільно рухатись вздовж станка, а різучий механізм – вздовж перекладки, покриваючи таким чином всю робочу поверхню. Типовий дизайн плоттера фірми Zünd [1] показано на рисунку 1.1.

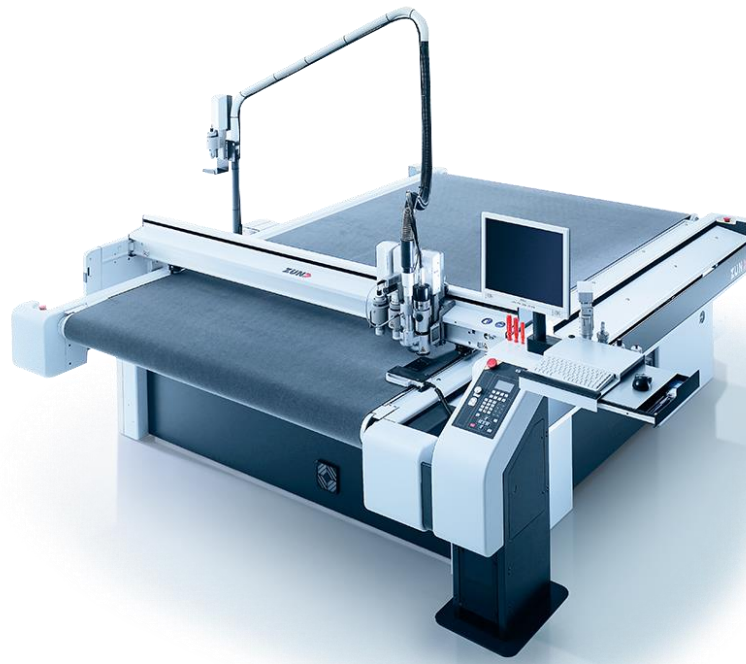


Рисунок 1.1 – Типова сучасна різуча система Zünd G3

Плоттери відрізняються за своїми розмірами, потужністю, цільовою областю застосування, універсальністю. Одна й та сама модель станка може випускатись в різних розмірах, зі змінними ріжучими насадками для різних типів матеріалів у залежності від потреб клієнта.

Зазвичай при оцінці плоттера прийнято у першу чергу звертати увагу на максимальну швидкість руху його ріжучого механізму. Це витікає з міркувань економії часу, адже швидший плоттер і нарізку виконає швидше. Але таке припущення часто буває помилковим. У специфікації частіше за все вказується максимальна швидкість та прискорення при русі різця вздовж прямої лінії в одному напрямі. Але на ділі цифри можуть дуже відрізнятись, враховуючи те, що вирізати часто доведеться часто змінюючи напрям руху різця уздовж кривих, а не прямих ліній. Різальний механізм також часто зупиняється у вершинах і кутах траєкторії, повертається та повторно вводить лезо в матеріал, що також впливає на загальну швидкість. При нарізці на швидкість руху різального механізму головним чином впливають матеріал та технічні властивості різця, тому плоттер з найвищими швидкісними показниками далеко не завжди буде найпродуктивнішим.

Слід пам'ятати, що продуктивність включає не тільки час, за який плоттер виконує нарізку, але й час підготовки та розкладання на різальній поверхні матеріалів нарізки, конфігурування програми та прибирання нарізаних об'єктів з робочої поверхні. Для серійної нарізки найкраще підходять станки з розширеною робочою поверхнею, які дозволяють збирати нарізані об'єкти, та розміщувати нові без переривання процесу нарізки, як це показано на рисунку 1.2. Без таких можливостей навіть найшвидший плоттер буде залежати від швидкості, з якою оператор прибирає нарізаний матеріал та розміщує новий. У контексті швидкодії, скорочення такого часу простою може бути в багато разів більш ефективним, ніж подвоєння швидкості руху ріжучого механізму.

Для визначення реальної швидкодії в конкретних умовах, клієнт повинен сам заміряти час, необхідний для нарізки декількох копій різних об'єктів з обраного ним матеріалу.

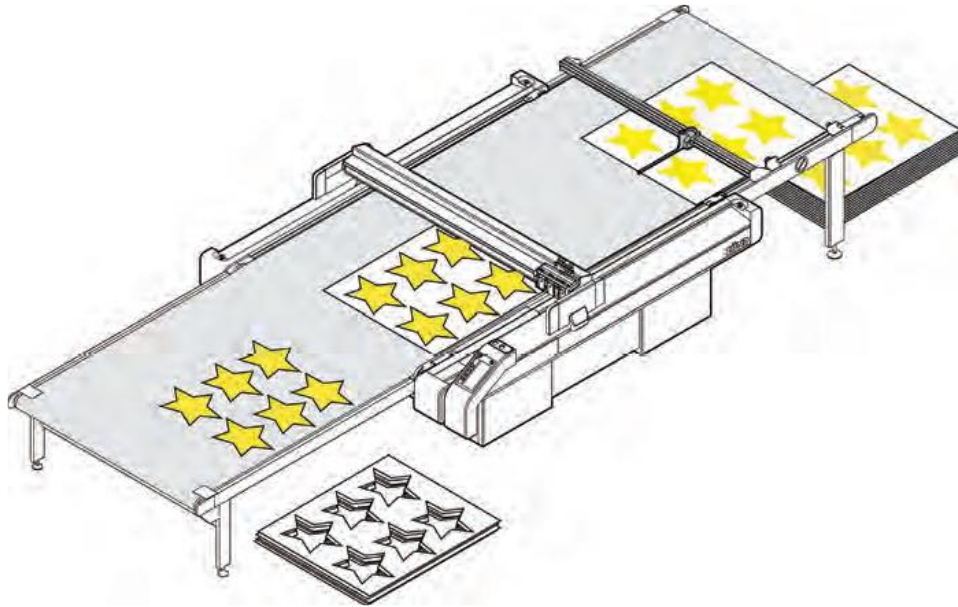


Рисунок 1.2 – Плоттер з розширеною робочою поверхнею

Нарізка з використанням двох незалежних різців одночасно, звичайно, подвоїть швидкодію плоттера.

Зазвичай вважається, що громіздкі та важкі на вигляд плоттери можуть різати товстіші матеріали швидше. Але, як жоден механізм не може бути міцнішим за свою найслабшу ланку, так і найміцніший на вигляд станок не буде міцнішим за свою найслабшу деталь.

У процесі різання реальна сила, що прикладається до леза різця в основному залежить від його форми та гостроти (його здатності різати), глибини розрізу та швидкості, з якою робиться розріз. Усі сучасні різці здатні легко розвивати таку потужність, яка просто пошкодить або зламає лезо. Тож більші двигуни або міцніші конструкції, здатні прикладати більші зусилля, не обов'язково дають очікувані переваги.

Вважається, що при нарізці легких, але міцних матеріалів, таких як надміцний картон Re-board або X-board, потрібне прикладення великого зусилля. Але для нарізки в даному випадку першочергового значення набуває вага матеріалу, а не його міцність і твердість. Менша вага тут означає, що матеріал містить більше повітря і ріжеться досить легко. Сучасні плоттери ріжуть матеріал Re-board зі швидкістю 1 м/с, прикладаючи всього 100 Ньютонів горизонтальної сили.

Ще однією цінною, але часто ігнорованою можливістю сучасних плоттерів, є здатність спрямовувати різець, зберігаючи потрібне заглиблення в матеріал відповідно до рельєфу його поверхні. Ця можливість дуже важлива при нарізці алюмінієвих композитних панелей, гравюр, шрифтів Брайля та в сфері виготовлення освітлюваних акрилових вітрин та написів, оскільки всі вони мають v-подібні заглиблення або пази, глибину яких потрібно враховувати при нарізці.

Деякі плоттери, тим не менш, здатні виконувати нарізку лише відносно своєї робочої поверхні. При такому режимі нарізки, відповідно до якості робочої поверхні станка, зазвичай забезпечується точність до десятої міліметра. Часто нарізка виконується у конвеєрному режимі, і тоді на якість нарізки додатково впливає конвеєрна стрічка. Захисне покриття, яке наносять на конвеєрну стрічку з метою уникнення її пошкодження при нарізці також вносить деяку незначну похибку. Додавши сюди ще кілька десятих міліметра від допуску товщини матеріалу що нарізується, отримуємо вже досить значну похибку при нарізці. Доволі часто можуть зустрічатись ситуації, коли частинки попередньо нарізаних матеріалів застрягають у проміжках між столом та матеріалом, і тоді підсумовуючи всі похибки, видно, що точна нарізка відносно різальної поверхні плоттера стає неможливою. Але, як вже було згадано вище, існують такі види різальних робіт, які потребують точного заглиблення різця і для яких потрібну точність може забезпечити тільки нарізка відносно поверхні матеріалу, а не станка.

Інші цінні, але маловідомі, можливості плоттерів включають нарізку м'яких сплавів та 3D-підписів, створення Брайлівських шрифтів, нарізка під різними кутами та можливість різати текстиль. Цифровий друк на текстилі займає найбільшу нішу у сфері цифрового друку, у той час як різання по кромці прінтів досі часто виконується вручну, спричиняючи значну затримку у виробництві. Цифрова нарізка прінтів на текстилі, тим не менш, досить складне завдання, яке стає ще більш складним при наявності зморшок або розтягнень текстилю. Деякі плоттери автоматизують цей процес, використовуючи розмотування закріпленого поруч рулону текстилю прямо на різальну поверхню, як це

показано на рисунку 1.3. Проте такий спосіб, усуваючи проблему зморщування, залишає відкритим питання розтягування матеріалу.



Рисунок 1.3 – Безперервна нарізка рулону текстилю

Під час різання таких матеріалів як текстиль, до різця прикладається зусилля, що далі передається матеріалу, тому необхідно переконатись у тому, що матеріал не буде деформуватись або зсуватись. Популярні плоттери використовують вакуумну систему фіксації матеріалу впродовж нарізки, але не всі вони працюють однаково ефективно.

Більшість систем використовують вакуумні помпи для утворення негативного тиску на матеріал. Такі помпи зазвичай оцінюються за їх номінальною потужністю і плоттер може мати в комплекті наприклад дві помпи потужністю 4 кВт кожна.

Проте це не завжди відображає реальну потужність помп та їх здатність утримувати матеріал на місці під час нарізки. Чи буде помпа утворювати негативний тиск на матеріал, значною мірою залежить від герметичності вакуумної системи і відсутності доступу повітря в неї. Це означає, що всі вільні зони вакуумного покриття мають бути накріті чимось непроникним для повітря для досягнення максимального ефекту. Але в

реальних умовах це майже неможливо. Наприклад, підчас нарізки завжди буде утворюватися розрив з доступом повітря уздовж лінії розрізу. І чим довша лінія розрізу, тим більшим буде потік повітря, і, як наслідок, меншим стане вакуумний ефект фіксації матеріалу. Так само при роботі з пористими матеріалами на основі паперу або текстилем повітря буде проходити крізь матеріал, утворюючи розгерметизацію вакуумної системи і знижуючи якість фіксації.

На практиці фіксація матеріалу залежить одночасно від утвореного помпою вакууму та від направленої потоку повітря. Системи, здатні утворювати потужний потік повітря для фіксації, зазвичай працюють краще, особливо при нарізці пористих матеріалів, де вакуумна система не може забезпечити потрібної якості фіксації. Ті системи, які не здатні створювати фіксуємого спрямованого потоку повітря, вимагають додаткових витрат часу на перекриття невикористаних зон вакуумного покриття та більших витрат енергії.

Більшість сучасних плоттерів працюють під управлінням програмного забезпечення різного ступеня складності. Але людське втручання і досі грає значну роль, адже власне нарізка – це найчастіше останній процес перед тим, як продукт буде запаковано та передано замовнику. Налаштування параметрів, підбір леза та його монтування, розкладання об'єктів нарізки на робочу поверхню та багато інших підготовчих процедур – це все зазвичай повинно виконуватись вручну оператором. Часто потрібне людське втручання і під час процесу нарізання, якщо щось пішло не за планом або нарізка поділяється на етапи, між якими потрібні зміни в обладнанні, програмі або розташуванні об'єктів.

Протягом свого функціонування у промислових масштабах рухомі деталі плоттера роблять величезну кількість рухів та покривають сумарну відстань у тисячі кілометрів, тому надійність є одним з ключових параметрів якісних станків для ЦНМ. Тим не менш, надійність може виражатись по-різному і не існує єдиної міри для неї в даному випадку. Зазвичай плоттер повинен мати значний запас міцності і ваги для забезпечення стабільної роботи в умовах частого прискорення та гальмування його рухомих деталей. При

цьому вага його рухомого устаткування має бути якомога нижчою, оскільки часте прискорення та гальмування важких деталей потребує вироблення та гасіння більшої кількості енергії, що призводить до значнішого зносу. Вібrazio є ще одним небезпечним для механічного та електротехнічного обладнання явищем, тому повинні бути зведені до мінімуму. Гладкість зрізу, наприклад, акрилу (рис. 1.4), може бути хорошим індикатором наявності значних вібрацій в системі.



Рисунок 1.4 – Зріз акрилової деталі за відсутності вібрацій в системі

Часто надійність виражають як час між відмовами в системі, маючи на увазі, як часто варто очікувати поламак техніки. У такому випадку, краще за все звертатись за детальною інформацією щодо статистики відмов при довгостроковому промисловому використанні плоттера безпосередньо до його виробника. Зазвичай великі бренди різальної індустрії, що турбуються про свій імідж, самі збирають і охоче надають таку статистику.

Маючи на увазі, що жодна система не міцніша за її найслабшу ланку, важливо щоб станок був оснащений лише оригінальними деталями найвищої якості. Найкращі леза та запчастини виробляються з максимальною точністю за найвищими стандартами якості як по відношенню до сировини, з якої вироблена деталь, так і до методів її обробки. Обертаючись зі швидкістю 30000 обертів на хвилину, низькоякісні деталі різця можуть зламатись самі та легко вивести його з ладу, навіть якщо всі інші деталі мають значно

вищу якість. Тому враховуючи сумарні витрати на операцію (матеріал, техніка, робочий час і т.д.), можливість зекономити на деталях може легко перетворитись на дуже дорогий урок.

Відповідальні виробники наносять унікальне маркування на кожне лезо для того, щоб відслідковувати будь-які відхилення від норми в функціонуванні та надійності різця. Це допомагає виробникам безперервно покращувати якість деталей.

## 1.2 Застосування комп'ютерного зору в сфері цифрової нарізки

Передові фірми світу, які займаються розробкою та випуском високоточної техніки завжди намагаються використовувати найсучасніші технології та вдосконалювати існуючі рішення за рахунок технічного прогресу. Так само і у сфері ЦНМ йде постійна інтеграція новітніх технологій, у тому числі і комп'ютерного зору. Тут впровадження систем комп'ютерного зору відбувалось крок за кроком, відповідно до ступеню розвитку техніки. Першим і головним кроком стало встановлення *локальної камери* прямо біля ріжучого механізму. Її «погляд» охоплює лише невелику ділянку різальної поверхні безпосередньо під різцем. Дані, отримані від камери після ретельного дослідження всієї різальної поверхні, дозволяють скласти точну карту розміщення об'єкта(-ів) на ній і визначити програму слідування різця по матеріалу згідно з заздалегідь визначеним шаблоном. Близькість до об'єктів на різальній поверхні забезпечила високу точність розпізнавання і позиціонування об'єктів, і, як наслідок, максимальну точність нарізки. Усі сучасні плоттери використовують дану технологію як базову.

Однак такий підхід має суттєвий недолік – швидкодію. Різальному механізму з камерою необхідно пройти вздовж усього контуру об'єкта нарізки, щоб скласти повну і точну картину для подальшого нарізання. Частково вирішити цю проблему дозволяє реконструкція позиції всього об'єкта, базуючись на отриманих даних про положення його частини. Утім економія часу за рахунок цього неминуче призводить до втрати точності, оскільки розрахунок ведеться на ідеально рівній, недеформований відносно шаблону об'єкт, чого інколи досить важко уникнути.



Інший підхід до рішення проблеми швидкодії полягає в нанесенні на нарізуваний матеріал опорних маркерів, так званих реєстраційних марок (далі реєстр. марок). Реєстр. марки являють собою чорні круги діаметром від 8 до 16 міліметрів. Набір реєстр. марок і їх взаємне розташування ідентифікують об'єкт і визначають спосіб його нарізки, як показано на рисунку 1.5. Такий спосіб є основним у сфері серійної нарізки, де на неперервному матеріалі (наприклад на рулоні текстилю, паперу і т.п.) можна розмістити довільну кількість промаркованих об'єктів для вирізання.

Тим не менш, навіть використання реєстр. марок не дозволяє повністю вирішити проблему швидкодії, так як камері перед виконанням нарізки так само треба виконати пошук марок на всій різальній поверхні. Ще одним серйозним недоліком такого підходу до пошуку об'єктів нарізки є подвійне зношування механізмів різачка, оскільки для виконання однієї операції нарізки йому необхідно двічі пройти вздовж різальної поверхні: перший раз для пошуку і позиціонування об'єктів на ній, а другий – для їх нарізки.

У даній дисертації мова піде про наступний етап технології ЦНМ, котрий полягає у застосуванні аналізу повного зображення всієї різальної поверхні і практично позбавлений недоліків локального пошуку об'єктів.

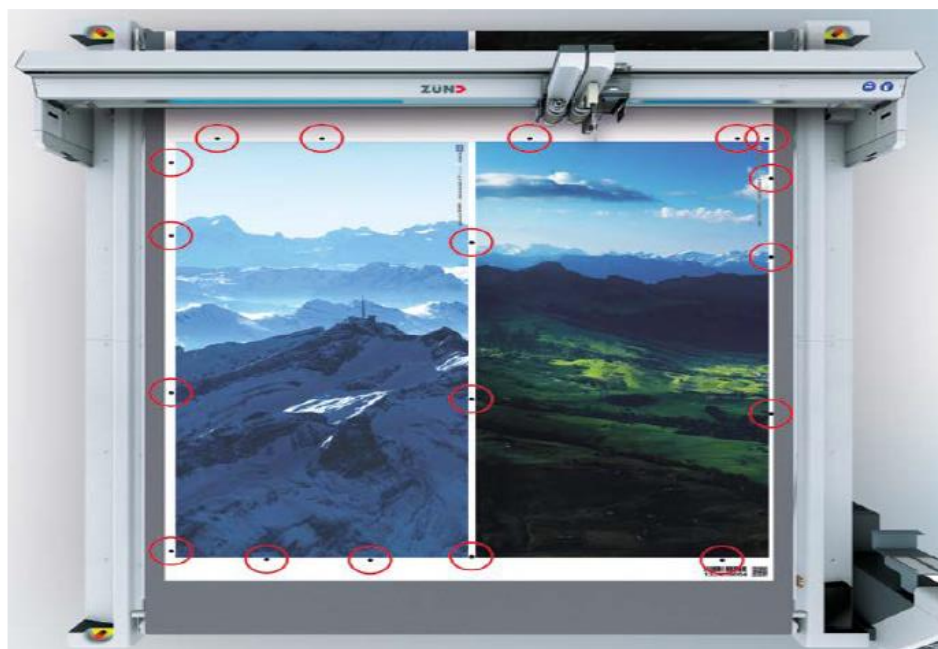


Рисунок 1.5 – Текстиль промаркований за допомогою реєстраційних марок

## 2 ЗАДАЧІ, МОДУЛІ ТА ІНСТРУМЕНТИ РОЗРОБЛЮВАНОЇ СИСТЕМИ

### 2.1 Постановка задачі та вхідні дані

Загальна задача описуваної системи комп'ютерного зору полягає в ідентифікації об'єктів на різальній поверхні плоттера за допомогою растрового зображення цієї поверхні разом з об'єктами нарізки на ній. Цільова система комп'ютерного зору (СКЗ) призначена виключно для роботи з об'єктами нарізки, що мають універсальний спосіб ідентифікації – реєстраційні марки. Крім фотографії різальної поверхні, системі також необхідна база даних шаблонів нарізки у вигляді файлів з координатами реєстр. марок у межах самого об'єкту для ідентифікації об'єктів знайдених на фотографії різальної поверхні.

У результаті описувана СКЗ повинна за мінімальний час, проаналізувавши зображення робочої поверхні різального станка, ідентифікувати всі розташовані на ній об'єкти, а також визначити їх положення відносно точки відліку різального механізму. Далі отримані дані передаються керуючій програмі різального станку у вигляді структур даних, що містять ідентифікатор об'єкта та координати всіх його реєстр. марок відносно заданої точки відліку.

Окремою задачею тут треба виділити створення системи позиціонування, яка б дозволяла знаходити точне положення реєстраційних марок в тій самій системі координат, яку використовує різальний механізм. Таку систему необхідно створювати перед подальшим використанням системи з тією самою конфігурацією камери та різальної поверхні. Також необхідно звернути увагу на усунення оптичних спотворень, які неминуче присутні навіть у найкращій сучасній фототехніці.

Ще одна підзадача полягає в проведенні обробки та ретельного аналізу зображення з метою якомога повного та точного пошуку реєстраційних марок на ньому. Після отримання всієї інформації про кількість та взаємне положення реєстраційних марок, можна переходити до визначення об'єктів нарізки серед них за відповідністю поперед-

ньо визначеним шаблонам. Останнім кроком треба виконати перехід від координат марок об'єктів на фотографії до координат у площині руху різального механізму та сформувати пакет даних для передачі керуючій програмі плоттера.

У підсумку, декомпозиція головної задачі функціонування розроблюваної програми виглядає наступним чином:

1) Створення сумісної системи позиціонування для ріжучого механізму плоттера один раз перед подальшим розпізнаванням об'єктів нарізки.

2) Усунення оптичних спотворень на фотографії.

3) Обробка зображення для збільшення точності розпізнавання.

4) Пошук реєстраційних марок на зображенні.

5) Ідентифікація окремих об'єктів нарізки серед знайдених реєстраційних марок за відповідністю до їх шаблонів.

6) Перехід від координат об'єктів на фотографії до координат відносно точки відліку різального механізму плоттера.

7) Формування пакету даних для передачі керуючій програмі плоттера.

## 2.2 Основні модулі розроблюваної системи

На основі визначених підзадач розроблюваної системи комп'ютерного зору можна визначити її основні підсистеми та модулі. Деякі задачі підлягають об'єднанню у межах окремого модулю в силу їх взаємної залежності або необхідності разового застосування.

В окремий модуль можна винести підсистему, що відповідає за позиціонування визначених на фотографії об'єктів відносно різального механізму плоттера. Для цього необхідно буде завжди робити окремий знімок різальної поверхні разом з розташованою на його поверхні координатною сіткою. Разом з визначенням системи позиціонування можна також виконати розрахунок корекції оптичних спотворень, притаманних використуваній камері та ракурсу, з якого робляться фото робочої поверхні станка – калібрування камери. За ці процедури буде відповідати модуль калібрування та позиціонування.

Обробка зображення перед подальшим розпізнаванням реєстраційних марок на ньому полягає у застосуванні до нього таких операторів, які підвищують чіткість контурів, збільшують контрастність та спрощають пошук реєстраційних марок на ньому. Для розпаралелювання процедури пошуку марок на зображенні, його можна розділити на сегменти меншого розміру, паралельна обробка яких як пришвидшить, так і покращить якість пошуку марок. Весь цей комплекс процедур можна винести в окремий модуль обробки зображення.

Однією з найбільш відповідальних процедур є пошук реєстраційних марок на зображенні. Для цього необхідно розробити надійний метод, який би максимально швидко знаходив та повертав координати у пікселях всіх наявних на поданому зображенні реєстраційних марок. При чому треба звести до мінімуму можливість повернення координат об'єктів, які за формою та розміром схожі на марку, але не є нею. Можливість появи таких об'єктів дуже висока, оскільки основним клієнтом у сфері цифрової нарізки є друкарська промисловість, тож поява малюнків з темними об'єктами округлої форми дуже вірогідна. Весь комплекс процедур, що займаються пошуком реєстраційних марок на фотографії формує ще один логічний модуль розроблюваної системи.

Усі попередні модулі у підсумку слугують для того, щоб надати якомога більш точну і вичерпну інформацію найбільш відповідальній задачі – пошуку об'єктів для нарізання серед знайдених реєстраційних марок. Оскільки шаблони подаються у формі списку координат реєстраційних марок від довільної точки відліку у міліметрах, а знайдені на знімку марки мають координати в пікселях відносно верхнього лівого кута знімку, необхідно буде звести їх до одної системи координат. Для цього знадобиться модуль калібрування і позиціонування. Лише після цього можна буде шукати співпадіння між конфігураціями знайдених на зображенні марок та шаблонами об'єктів нарізки. Ця задача нагадує пошук сузір'я на зоряному небі, де в ролі зірок виступають знайдені на фото марки, а форму шуканого сузір'я задає шаблон об'єкта нарізки. Цей алгоритм буде реалізовано в межах модуля ідентифікації об'єктів нарізки.

Усі процедури, пов'язані з прийомом та передачею пакетів даних між цільовою системою комп'ютерного зору та керуючою програмою плоттера, також можна винести в окремий модуль. Проте в силу різноманітності можливих протоколів обміну, які застосовують різні фірми-розробники плоттерів, його реалізація може різнитись від одного до іншого типу керуючого ПЗ плоттера. Особливої уваги останньому модулю приділено не буде, оскільки темою даної дисертації є розробка якомога більш універсальної концепції системи комп'ютерного зору для сфери ЦНМ, незалежно від особливостей устаткування тої чи іншої фірми.

## 2.3 Технічна база та інструменти розробки системи комп'ютерного зору

### 2.3.1 Необхідна технічна база розроблюваної системи

Для забезпечення технічної бази описуваної системи комп'ютерного зору необхідне встановлення додаткової *глобальної фотокамери* над різальною поверхнею плоттера. З метою отримання оптимального для аналізу зображення, камеру слід розташувати над центром різальної поверхні так, щоб оптична вісь була перпендикулярна до неї. Висоту треба обрати достатню для отримання зображення всієї робочої поверхні станка з мінімумом зайвих об'єктів на фото.

Нерухоме закріплення камери є ключовим моментом, оскільки через можливі вібрації плоттера під час роботи, у разі зміщення її оптичної осі, усі наступні, зроблені нею, знімки не будуть відповідати виробленій системі позиціонування відносно різального механізму, зробленій до зміщення. Чим вища роздільність та якість знімків камери, тим кращою буде точність позиціонування і ідентифікації об'єктів на них, що свідчить на користь використання якомога більш якісних фотокамер для даної цілі.

Ще одним важливим моментом є наявність повноцінного програмного інтерфейсу для керування камерою за допомогою ПЗ, а не вручну. Оскільки особливості монтування камери сильно залежать від моделі плоттера та доцільності її використання для конкретної моделі, яка може бути розроблена для потреб, які не передбачають використання глобальної камери, відповідальність за її монтаж та включення в специфікацію плоттера

цілком покладається на фірму, яка буде впроваджувати цільову технологію в свою продукцію.

Оскільки всі сучасні плоттери оснащені комп'ютером за замовчуванням, вимога наявності необхідної обчислювальної техніки для запуску програми комп'ютерного зору задовольняється заздалегідь. Операційна система тут значення не має, оскільки ядро цільової системи є повністю незалежним від платформи, як і всі бібліотеки, використані під час його розробки. Продуктивність програми буде цілком залежати від потужності обчислювальної техніки, на якій вона розгортається.

### 2.3.2 Обґрунтування вибору C++ в якості мови програмування для розробки ядра системи

У якості мови програмування для створення програмної частини системи комп'ютерного зору обрано мову C++, оскільки вона якнайкраще підходить для розробки кросплатформених програмних продуктів з високими вимогами до швидкодії та споживання пам'яті.

C++ (Сі-плюс-плюс) — мова програмування високого рівня з підтримкою кількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Розроблена Б'ярном Страуструпом в AT&T Bell Laboratories 1979 року. Базується на мові С. Уперше описана стандартом ISO/IEC 14882:1998, найбільш актуальним наразі є стандарт ISO/IEC 14882:2017. [2]

У 1990-х роках C++ стала однією з найуживаніших мов програмування загального призначення. Мову використовують для системного програмування, розробки програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм, наприклад, відеоігор. C++ суттєво вплинула на інші популярні сьогодні мови програмування: C# та Java.

Стандартна бібліотека C++ включає стандартну бібліотеку С з невеликими змінами, які роблять її відповідною для мови C++. Інша велика частина бібліотеки C++ за-

снована на Стандартній Бібліотеці Шаблонів (STL). Вона надає такі важливі інструменти, як контейнери (наприклад, вектори і списки) й ітератори (узагальнені вказівники), що надають доступ до цих контейнерів як до масивів. Крім того, STL дозволяє схожим чином працювати і з іншими типами контейнерів, наприклад, асоціативними списками, стеками, чергами.

Ключові переваги мови C++ над іншими C-подібними мовами програмування полягають в наступному:

- Швидкодія.

Швидкість роботи програм на C++ практично не поступається програмам на C, хоча програмісти отримали у своїх руках нові можливості і нові засоби. У нашому випадку це є необхідною умовою реалізації програми, яка однією зі своїх цілей ставить економію часу оператора. Особливо ця перевага буде проявлятися під час виконання складних алгоритмів узгодження множин точок, які виконують велику кількість математичних розрахунків.

- Масштабованість.

На мові C++ розробляють програми для найрізноманітніших платформ і систем. Для нас це дуже важливий критерій, оскільки справа стосується дуже специфічної галузі, де розроблювану СКЗ доведеться застосовувати з найрізноманітнішими системами та платформами, які змінюються від одного виду плоттера до іншого.

- Можливість роботи на низькому рівні з пам'яттю, адресами, портами. (Що при необережному використанні може легко перетворитися на недолік.)

Буде корисною перевагою при взаємодії з інтерфейсами камери та плоттера, адже вона відбувається на низькому рівні, максимально близькому до електроніки.

- Можливість створення узагальнених алгоритмів для різних типів даних, їхня спеціалізація і обчислення на етапі компіляції, з використанням шаблонів.

Це буде перевагою при побудові алгоритмів розпізнавання та створенні гнучкої модульної архітектури ядра системи. Використання будь-якої іншої мови призвело б у підсумку до менш зв'язної та логічної архітектури.

- Підтримуються різні стилі та технології програмування, включаючи традиційне директивне програмування, ООП, узагальнене програмування, метапрограмування (шаблони, макроси).

Можливість змінювати технологію використана при роботі з різними фреймворками та бібліотеками. Наприклад, QtFramework націлений на ООП, у той час як OpenCV передбачає використання функціонального стилю програмування.

До недоліків C++ можна віднести наступні моменти:

- Наявність безліч можливостей, що порушують принципи безпеки типів, приводить до того, що в C++ програми може легко закрастися важко відстежувана помилка. Замість контролю з боку компілятора, розробники вимушені дотримуватися вельми нетривіальних правил кодування. По суті ці правила обмежують C++ рамками якоїсь безпечнішої підмови. Більшість проблем безпеки типів C++ успадкована від C, але важливу роль у цьому питанні грає і відмова автора мови від ідеї використовувати автоматичне управління пам'яттю (наприклад, збірку сміття). Так візитною карткою C++ стали вразливості типу «переповнювання буфера».

У той же час сучасні стандарти C++ надають багато інструментів для уникнення вказаних проблем, як, наприклад, «інтелектуальні» вказівники, які самі виконують очищення пам'яті, звільняючи програмістів від зайвих клопотів. При розробці СКЗ такі інструменти використовувались усюди, де була така можливість.

- Погана підтримка модульності. Підключення інтерфейсу зовнішнього модуля через препроцесорну вставку заголовного файлу (`#include`) серйозно уповільнює компіляцію при підключенні великої кількості модулів. Для усунення цього недоліку багато компіляторів реалізують механізм попередньої компіляції заголовних файлів (англ. Precompiled Headers).

Цього недоліку вдалося уникнути за рахунок використання досконалого компілятора від Microsoft Visual Studio.



### 2.3.3 Використання OpenCV як джерела потужних засобів комп'ютерного зору

Реалізація основних алгоритмів обробки та аналізу растрових зображень уже є у відкритій бібліотеці OpenCV (англ. Open Source Computer Vision Library), яку може вільно використовувати в академічних та комерційних цілях. Саме це робить OpenCV очевидним рішенням для розробників, які мають потребу у засобах комп'ютерного зору, але не мають можливості платити за комерційні бібліотеки з закритим кодом. Ось чому алгоритми цільової програми повністю базуються на використанні даної бібліотеки.

OpenCV (англ. Open Source Computer Vision Library, бібліотека комп'ютерного зору з відкритим кодом) — бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. Бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях (наприклад, осіб і фігур людей, тексту тощо), відстежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних зображеннях.

Починаючи з перших етапів розвитку OpenCV, основними задачами бібліотеки були:

- розвивати дослідження у напрямку комп'ютерного зору, забезпечуючи добре оптимізований та відкритий код бібліотеки.
- поширювати знання у сфері комп'ютерного зору, забезпечуючи загальну інфраструктуру, яку б могли розвивати розробники, таким чином код ставатиме більш легким для сприйняття та обміну.
- розвивати засновані на роботі з комп'ютерним зором комерційні додатки, створюючи незалежну від платформи, оптимізовану та безкоштовну бібліотеку. Для цього використовувалася ліцензія, яка не вимагала від таких комерційних додатків бути відкритими.

Бібліотека містить понад 2500 оптимізованих алгоритмів, серед яких повний набір як класичних, так і практичних алгоритмів машинного навчання і комп'ютерного зору.

У розроблюваній системі були використані алгоритми калібрування камери, пошуку за зразком та багато інших засобів, які в іншому випадку довелося б шукати в різних бібліотеках.

Алгоритми OpenCV застосовують у таких сферах:

- Аналіз та обробка зображень
- Системи з розпізнавання обличчя (рис. 2.1)
- Ідентифікації об'єктів
- Розпізнавання жестів на відео
- Відстежування переміщення камери
- Побудова 3D моделей об'єктів
- Створення 3D хмар точок зі стерео камер
- Склеювання зображень між собою, для створення зображень всієї сцени з високою роздільною здатністю
- Система взаємодії людини з комп'ютером
- Пошуку схожих зображень із бази даних
- Усування ефекту червоних очей при фотозйомці зі спалахом
- Стеження за рухом очей
- Аналіз руху
- Ідентифікація об'єктів
- Сегментація зображення
- Трекінг відео
- Розпізнавання елементів оточення і додавання маркерів для створення доповненої реальності

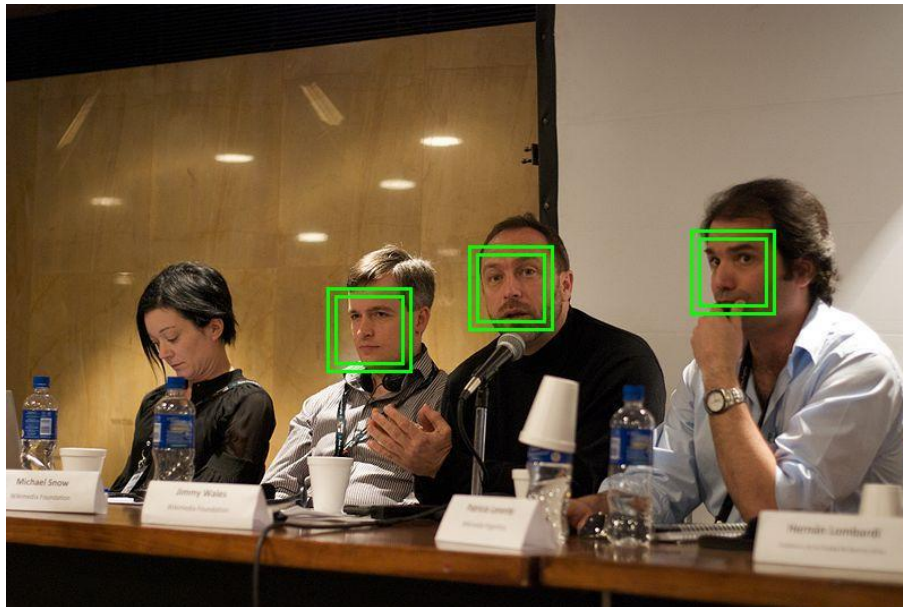


Рисунок 2.1 – Приклад автоматичного розпізнавання об'єктів засобами OpenCV

Спільнота OpenCV налічує понад 47 тисяч користувачів, а загальна кількість завантажень оцінюється в понад 14 мільйонів. Бібліотека широко використовується у сфері комерційної розробки програмного забезпечення, дослідницьких групах та урядових структурах. Цей факт не один раз ставав у нагоді, коли виникали труднощі під час упровадження того чи іншого функціоналу, оскільки в Інтернеті легко можна було знайти багато готових рішень на основі досвіду інших користувачів.

Поряд із добре відомими компаніями, такими як Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, OpenCV широко використовують багато нових компаній, таких як: Applied Minds, VideoSurf та Zeitera.

Діапазон використання OpenCV включає поєднання об'ємних зображень вулиць в єдине (Google Maps), виявлення вторгнень сторонніх осіб через відеоспостереження в Ізраїлі, моніторинг обладнання шахт в Китаї, допомоги роботам в навігації та розпізнаванні об'єктів в Willow Garage, виявлення потопаючих в басейнах Європи, перевірку злітно-посадкових смуг на предмет засмічення в Туреччині, перевірку етикеток продукції на заводах у всьому світі та для швидкого розпізнавання обличчя в Японії.

Усе це свідчить про OpenCV, як про бібліотеку, яка навряд чи втратить широку технічну підтримку у найближчі роки, а навпаки – продовжить стрімко розвиватись, допомагаючи розвивати і створену з її використанням СКЗ зараз. Так відпадає небезпека застарівання розроблюваної технології.

OpenCV має інтерфейси для мов програмування C ++, Python, Java та MATLAB і підтримує такі операційні системи, як Windows, Linux, Android та Mac OS. API для цих інтерфейсів можна знайти в онлайн документації [3]. У даний час активно розвиваються повнофункціональні інтерфейси CUDA та OpenCL. Існує понад 500 алгоритмів і приблизно в 10 разів більше функцій, які складають або підтримують ці алгоритми. В іншому випадку розроблена СКЗ була б вузько орієнтована, спеціалізована лише під конкретного виробника та інтерфейси, впроваджені в його техніку.

Хоча OpenCV написана на C++ і її основний інтерфейс реалізовано для цієї мови програмування, бібліотека і досі підтримує старіший C інтерфейс. Усі нові розробки та алгоритми OpenCV у даний момент розробляються спочатку для C++ інтерфейсу, а вже потім переносяться на інші.

#### 2.3.4 Можливості використання апаратного прискорення та паралельних обчислень з алгоритмами OpenCV

Сучасні графічні процесори стали потужними обчислювальними одиницями і мають достатньо можливостей для виконання загальних обчислень (GPGPU). Це дуже швидко зростаюча область, яка викликає великий інтерес вчених, дослідників та інженерів, які розробляють високоефективне програмне забезпечення. Незважаючи на труднощі реалізації алгоритмів на графічному процесорі, багато людей роблять це, щоб перевірити, наскільки швидко вони можуть виконуватись. Для підтримки таких зусиль було створено безліч передових мов та інструментів, таких як CUDA, OpenCL, C++ AMP, дебагери, профайлери та ін.

Значна частина комп'ютерного зору – це обробка зображень, тобто область, до якої використання графічних процесорів має безпосереднє відношення. Інші галузі також передбачають застосування паралельних обчислень і часто штучно переносяться на архітектуру графічних процесорів. Так і виникла ідея реалізувати всі ці переваги та прискорити OpenCV на графічних процесорах.

OpenCV включає модуль, що цілком базується на використанні можливостей графічного процесора і складається з алгоритмів, продуктивність яких значно вища за рахунок використання апаратного прискорення. За підтримки NVIDIA, робота над модулем розпочалась у 2010 році і продовжувалась до першого випуску навесні 2011 року. Він включає в себе прискорений код для значної частини бібліотеки і все ще продовжує розширюватись та адаптуватись до нових обчислювальних технологій та архітектур графічних процесорів NVIDIA.

Ця бібліотека була створена для того, щоб:

- 1) Надати розробникам зручний фреймворк комп'ютерного зору, заснований на використанні передових можливостей графічних процесорів.
- 2) Досягнути максимальної продуктивності завдяки графічним процесорам (ефективні ядра, налаштовані на сучасні архітектури, оптимізовану обробку даних, як, наприклад, асинхронне виконання, перехресне копіювання).
- 3) Повнота (переписування під графічні процесори якомога більшого функціоналу, навіть якщо приріст швидкості не є значним, але це дозволяє запускати алгоритм повністю на графічному процесорі та заощаджувати на скороченні кількості операцій копіювання).

На рисунку 2.2 наведено порівняльну статистику швидкодії деяких алгоритмів роботи із зображеннями при використанні графічного модуля Tesla C2050 проти процесора Core i5-760 2.8Ghz з використанням технологій SSE і TBB [4]:

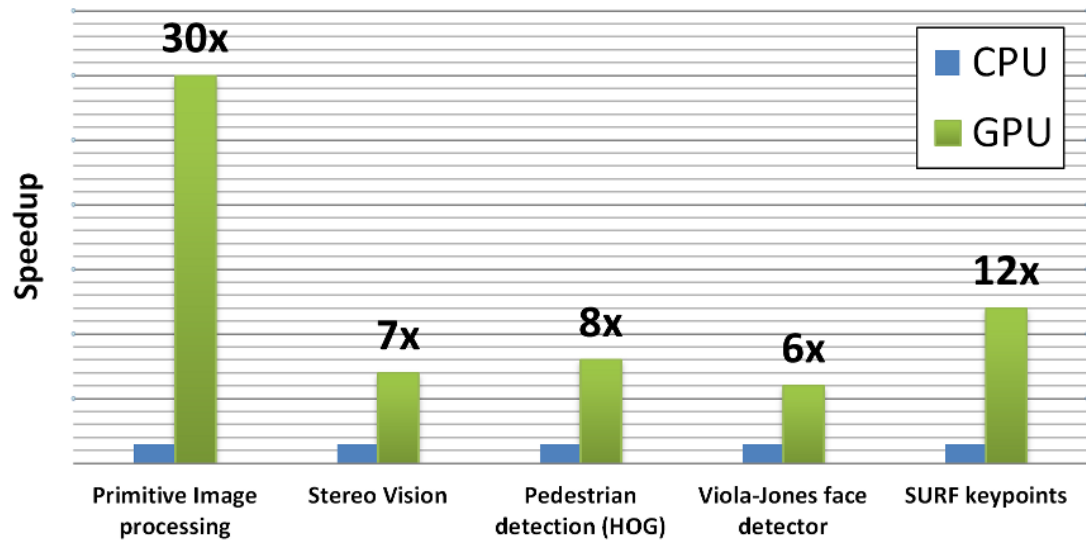


Рисунок 2.2 – Порівняння швидкодії сучасних центрального та графічного процесорів

Модуль графічного процесора OpenCV написано з використанням технології CUDA, тому на даний момент він може використовувати лише її можливості. Весь модуль був розроблений як розширення програмного інтерфейсу OpenCV. Ця надбудова надає користувачеві явний контроль над тим, як дані переносяться між пам'яттю ЦП і графічного процесора. Незважаючи на те, що користувач повинен написати додатковий код для початку використання графічного процесора, цей підхід є гнучким та дозволяє проводити більш ефективні обчислення.

Основу модуля складає клас `cv::gpu::GpuMat`, який є основним контейнером для даних, що зберігаються у пам'яті графічного процесора. Інтерфейс дуже схожий з `cv::Mat`, його альтернативою для ЦП. Усі функції, пов'язані з апаратним прискоренням, отримують `GpuMat` як аргументи введення та виведення. Це дозволяє викликати декілька алгоритмів без завантаження даних. Програмний інтерфейс модуля збережено максимально подібним до стандартного інтерфейсу OpenCV для ЦП. Тому розробники, які знайомі зі стандартним OpenCV, можуть одразу почати використовувати апаратне прискорення.

Додаткові можливості з використання апаратного прискорення в OpenCV закладено в технології OpenCL. Open Computing Language (OpenCL) – це відкритий стандарт

для написання коду, що може виконуватись на неоднорідних платформах, включаючи процесори, графічні процесори, процесори цифрової обробки сигналів і т. д. Зокрема, OpenCL надає програмам доступ до графічних процесорів для неграфічних обчислень (GPGPU), які в деяких випадках дають значний приріст у швидкості. У Computer Vision багато алгоритмів можуть працювати на GPU значно ефективніше, ніж на центральному процесорі, наприклад: обробка зображень, матрична арифметика, виявлення об'єктів тощо.

Інтеграція OpenCL в OpenCV розпочалося в 2011 році за ініціативою AMD. У результаті до випуску OpenCV-2.4.3 був включений новий модуль ocl, що містить OpenCL реалізацію деяких існуючих алгоритмів OpenCV. Тобто, коли на клієнтській машині є доступ до середовища OpenCL і сумісного пристрою, користувач може замість коду `cv::resize()` використовувати конструкцію `cv::ocl::resize()`, щоб скористатися прискореним кодом. Протягом наступних років все більше і більше функцій і класів було додано до модуля ocl, але це був окремий програмний інтерфейс поряд з основним процесорно-орієнтованим інтерфейсом в OpenCV-2.x.

В OpenCV-3.x концепція архітектури була змінена на так званий «Прозорий програмний інтерфейс (T-API)». У новій архітектурі окремий OpenCL-прискорений `cv::ocl::resize()` видаляється з зовнішнього API і стає відділенням у звичайному `cv::resize()`. Ця гілка виконується автоматично, коли це можливо і має сенс з точки зору продуктивності. Реалізація T-API була ініційована компаніями AMD та Intel.

Показники приросту продуктивності з використанням OpenCL проти стандартних обчислень за допомогою ЦП у версії OpenCV-3.0 наведено на рисунку 2.3.

У підсумку, усі ці засоби прискорення стають дуже важливими в системі, яка за своєю концепцією націлена на граничну продуктивність, аби забезпечити максимальний приріст швидкості роботи плоттерів. Компонування бібліотек відбувалось з увімкненою підтримкою усіх можливих засобів апаратного прискорення, що означає їх неодмінне використання на пристроях, які підтримують такі можливості, без внесення змін у програмний код.

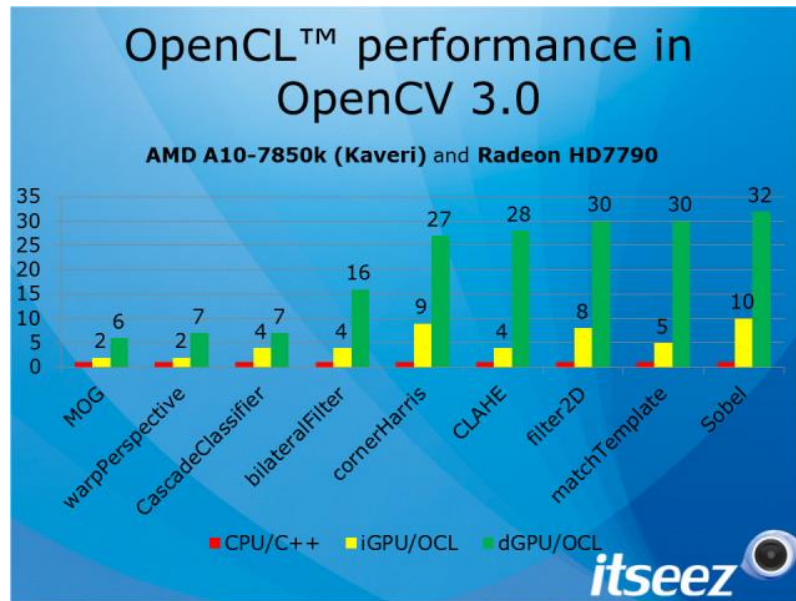


Рисунок 2.3 – Порівняння швидкодії стандартних алгоритмів OpenCV та їх OpenCL варіантів

### 2.3.5 Можливості Qt-фреймворку для побудови кроссплатформового ядра та графічного інтерфейсу

Qt — кроссплатформовий інструментарій розробки програмного забезпечення (ПЗ) мовою програмування C++. Дозволяє запускати написане за його допомогою ПЗ на більшості сучасних операційних систем (ОС), просто компілюючи текст програми для кожної операційної системи без зміни вихідного коду. Містить усі основні класи, які можуть бути потрібні для розробки прикладного програмного забезпечення, починаючи з елементів графічного інтерфейсу й закінчуючи класами для роботи з мережею, базами даних, OpenGL, SVG і XML. Бібліотека дозволяє керувати потоками, працювати з мережею та забезпечує кроссплатформовий доступ до файлів.

Усі ці переваги Qt переносяться на розроблювану систему, залишаючи її сумісною з більшістю сучасних платформ, та доповнюючи функціонал зручними засобами, на розробку яких шкода витратити час. Весь графічний інтерфейс демонстраційної програми створено засобами Qt, як і всі апаратнозалежні модулі, що підтримує повну незалежність розроблюваної системи від платформи.



Починаючи з версії 4.5 Qt поширюється за трьома ліцензіями (незалежно від ліцензії, сирцевий код Qt один і той же):

- Qt Commercial — для розробки ПЗ з власницькою ліцензією, що допускає модифікацію самої Qt без розкриття змін;
- GNU GPL — для розробки відкритого програмного забезпечення, що поширюється на умовах GNU GPL;
- GNU LGPL — для розробки ПЗ з власницькою ліцензією, але без внесення змін до Qt.

При розробці СКЗ була використана ліцензія GNU LGPL, оскільки створювана технологія не передбачається для вільного поширення та відкриття сирцевого коду.

З часу своєї появи в 1996 році комерційна версія бібліотеки Qt лягла в основу тисяч успішних проєктів у всьому світі. Крім того, Qt є фундаментом популярного робочого середовища KDE, що входить до складу багатьох дистрибутивів GNU/Linux.

Серед відомих проєктів особливо треба відзначити:

- програма для IP-телефонії Skype;
- програма для обробки зображень Adobe Photoshop Album;
- мережева карта світу Google Earth.

Так само як і у випадку з OpenCV, це гарантує те, що програма не застаріє у найближчі роки.

Відмінна особливість Qt від інших бібліотек — використання Meta Object Compiler (МОС) — попередньої системи обробки вихідного коду (загалом, Qt — це бібліотека не для чистого C++, а для його особливого діалекту, з якого й «перекладає» МОС для подальшої компіляції будь-яким стандартним C++ компілятором). МОС дозволяє в багато разів збільшити потужність бібліотек, вводячи такі поняття, як слоти (slots) і сигнали (signals). Qt комплектується візуальним середовищем розробки графічного інтерфейсу Qt Designer, що дозволяє створювати діалоги і форми «мишею».

Ідеологія створення форм у Qt базується на використанні менеджерів розташування, котрі надають гнучкий дизайн, при якому розмір і розташування елементів форм

визначаються автоматично, що значно прискорює розробку графічного інтерфейсу. У поставці Qt є «Qt Linguist» — могутня графічна утиліта, що дозволяє спростити локалізацію й переклад вашої програми багатьма мовами, та «Qt Assistant» — довідкова система Qt, що спрощує роботу з документацією для бібліотек і дозволяє створювати крос-платформову довідку для ПЗ, розробленого на основі Qt. Усе це може стати у нагоді при переході розроблюваної системи на комерційну основу, для підтримки різних мов та необхідності створення документації за міжнародними стандартами.

### 2.3.6 Microsoft Visual Studio 2015 як середовище розробки ядра та демонстраційної версії програми

Для створення програмних складових розроблюваної системи була використана Microsoft Visual Studio.

Microsoft Visual Studio – лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, є тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight.

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторінга коду. Вбудований відлагоджувач може працювати як для рівня вихідного коду, так і для машинного рівня. Решта вбудованих інструментів включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (як, наприклад, Subversion і Visual SourceSafe),

додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування) або інструментів для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server).

Обрана версія Visual Studio 2015, під кодовим ім'ям Dev14, була представлена 20 червня 2015 року. Суттєвою зміною стала підтримка багатьох цільових платформ: окрім базової Windows з'явилась можливість будувати проекти для IOS та Android. Для розробників комп'ютерних ігор була додана підтримка фреймворку Unity. Був оновлений механізм автентифікації: користувач під час запуску Visual Studio синхронізується з єдиним акаунтом Microsoft.

Версія включає в собі .NET Framework 4.6 та підтримку універсальної платформи Windows 10. Розробники на мові C++ отримали в розпорядження новий функціонал стандарту C++14 та навіть деякі поліпшення з C++17.

Останнім оновленням на даний момент є Update 2 від 30 березня 2016-го року, у якому багато уваги приділено стабільності та продовжено роботу у напрямку підтримки нових стандартів мови C++.

Ще однією перевагою є наявність розроблених спеціально для неї Qt Visual Studio Tools – інструментів, які інтегруються в Visual Studio 2015 та дозволяють використовувати в ньому засоби Qt, не витрачаючи часу на деякі моменти, властиві розробці та зборці програм з використанням Qt. Завдяки цьому додатку у середовищі Visual Studio 2015 стають доступними такі можливості з Qt Creator, як:

- Майстри для створення нових проектів та класів Qt.
- Автоматичне налаштування для компілятора мета-об'єкта (moc), компілятора користувачького інтерфейсу (uic) та компілятора ресурсів (rcc).
- Імпорт та експорт файлів проекту Qt (.pro) та включених до нього файлів (.pri).
- Автоматична конвертація проекту Qt VS Tools в проект qmake або навпаки.
- Інтегроване управління ресурсами Qt.
- Інтегрована документація Qt.

- Налаштування розширень для типів даних Qt.

Усі ці переваги Visual Studio 2015 стали у нагоді при розробці системи, яка поєднує у собі принципово різні технології (OpenCV, Qt), які зазвичай важко суміщаються без додаткових засобів.

## 3 КАЛІБРУВАННЯ І ПОЗИЦІОНУВАННЯ ОБ'ЄКТІВ НАРІЗКИ

### 3.1 Оптичні дисторсії та їх вплив на зображення різальної поверхні плоттера

Перед початком аналізу зображення різальної поверхні слід переконатись, що воно повністю відображає реальність. Отримані за допомогою фотозйомки зображення часто «страждають» від різних оптичних спотворень (дисторсії) [5], вплив яких варіюється в залежності від якості оптики об'єктива камери.

Дисторсія (від лат. *distorsio, distortio* — викривлення) — аберация оптичних систем, при якій лінійне збільшення змінюється по полю зору, порушуючи подібність між об'єктом та його зображенням. Оптична система, вільна від дисторсії, називається ортоскопічною, оскільки задовольняє вимогам ортоскопічності [6].

«Подушкоподібна» дисторсія виникає у випадку, коли лінійне збільшення зростає з віддаленням елементів зображення від оптичної осі. Візуально це виглядає так, що лінії, які не йдуть через центр зображення, вигинаються всередину у напрямку до центру зображення як подушка.

Коли лінійне збільшення зменшується з віддаленням елементів зображення від оптичної осі — виникає «бочкоподібна» дисторсія. Візуальний ефект є таким, ніби зображення нанесене довкола сфери (або бочки). Лінзи «риб'яче око», які мають напівсферичне поле зору, використовують цей тип дисторсії як спосіб відображення нескінченно широкої площини об'єкта у скінченну площу зображення. В об'єктивах бочкоподібна дисторсія з'являється в центрі діапазону фокусних відстаней об'єктиву і стає більшою в широкому куті діапазону.

Поєднанням обох типів іноді називається вусоподібною дисторсією або складною дисторсією. Вона починається як бочкоподібна дисторсія ближче до центру зображення і плавно переходить у подушкоподібну дисторсію в напрямок до краю зображення так, що горизонтальні лінії у верхній половині кадру виглядають як вуса. Вплив обох видів дисторсії на вихідне зображення, а також принцип її оптичного виправлення можна побачити на рисунку 3.1.

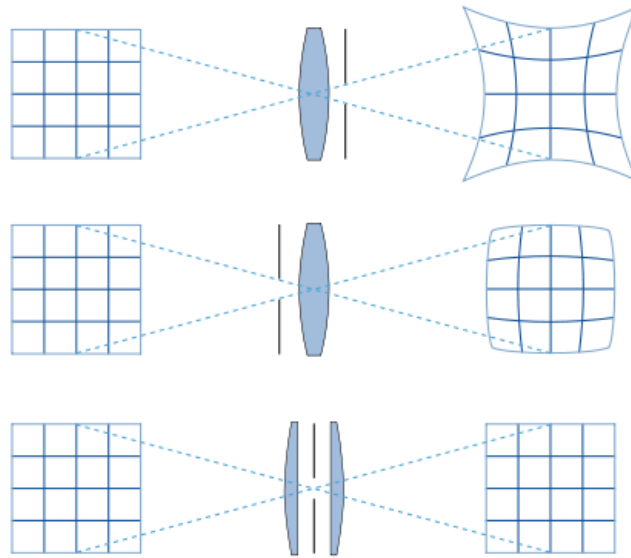


Рисунок 3.1 – Дисторсія зображення відповідно до типів лінз

Математично бочкоподібна і подушкоподібна дисторсія квадратичні. Це означає, що вони збільшуються із квадратом відстані від центру. У вусоподібній дисторсії квадратичний (ступінь 4) терм є значимим: у центрі, ступінь 2 бочкової дисторсії буде переважати, у той час як на краях буде переважати ступінь 4 подушкоподібної дисторсії. Інші дисторсії також можуть бути можливі в принципі – подушкоподібна в центрі і бочкоподібна на краях або дисторсія більш високого порядку (ступеня 6, 8) – але зазвичай вони не трапляються на практиці в об'єктивах, і дисторсії вищого порядку зазвичай мають малий зв'язок саме з бочкоподібним або подушкоподібним ефектами.

В окремих випадках спотворення форми можуть мати і більш складний вид. Застосування апертурної або віньєтуючої діафрагми не впливає на дисторсію, оскільки така діафрагма не змінює коефіцієнт збільшення оптичної системи.

Дисторсія може бути виражена кількісно через так звану відносну дисторсію  $v$  за формулою:

$$v = \left[ \frac{(b-b_0)}{b} \right] \cdot 100 \% \quad (3.1)$$

де  $b_0$  – коефіцієнт лінійного збільшення ідеальної системи (системи без дисторсії);  
 $b$  – дійсний коефіцієнт збільшення, дві безрозмірні величини.

Як видно з формули відносна дисторсія  $v$  вимірюється у відсотках.

Коефіцієнт збільшення  $b$  на оптичній осі дорівнює  $b_0$ . Відхилення від  $b_0$ , зазвичай, досягає максимуму по краю поля зору. Тому для характеристики дисторсії оптичної системи зазвичай за величину  $b$  приймають коефіцієнт збільшення по краю.

Для однієї і тієї ж системи дисторсія залежить:

- від відстані до об'єкта;
- від коефіцієнта збільшення  $b_0$ ;
- від довжини хвилі.

Як правило, якщо дисторсія мала або відсутня при одній відстані, вона буде мала і при іншій.

У фотографії дисторсія зазвичай асоціюється із трансфокаторами – об'єктивами зі змінною фокусною відстанню, особливо при збільшенні в широкому діапазоні, але також зустрічається і в об'єктивах з фіксованим фокусом, і залежить від фокусної відстані – наприклад, об'єктив Canon EF 50mm f/1.4 породжує бочкоподібну дисторсію на дуже малих фокальних відстанях. Бочкоподібна дисторсія зустрічається в ширококутних об'єктивах і, зазвичай, спостерігається на широких кутах огляду у трансфокаторах, у той час як подушкоподібна дисторсія зазвичай спостерігається у більш старих або в телеоб'єктивах низької якості. Складна дисторсія зустрічається зокрема на широких кутах деяких трансфокаторів, у ретрофокусних об'єктивах і в більш недавніх моделях з широким діапазоном, таких як Nikon 18–200 mm.

Для того щоб зрозуміти природу дисторсії, слід пам'ятати що це радіальні викривлення. Оптичні системи в більшості мають обертальну симетрію (якщо не брати до уваги нерадіальні дефекти), таким чином дидактично правильне тестове зображення буде мати концентричні кола, що мають однакові поділки — як ціль для стрільби. На ньому можна буде спостерігати, що ці загальні дисторсії мають нелінійне відображення радіусів на зображення, що здавалося б є подушкоподібним спотворенням, насправді лише збільшує радіуси кіл при відображенні для більших радіусів в порівнянні з малими

радіусами. Графік, який показує перетворення радіусів (від реального об'єкта на отриманому зображенні) буде крутішим у верхньому (правому) кінці. І навпаки, бочкоподібна дисторсія насправді зменшує радіуси великих кіл в порівнянні з колами малого радіусу. Графік, який показує перетворення радіусів буде менш крутим в верхній (правій) частині.

Радіальна дисторсія, яка залежить від довжини світлової хвилі, називається «бічною хроматичною аберациєю» — «бічна» тому що радіальна, «хроматична» тому що залежить від кольору (довжини хвилі). Це може призвести до появи кольорових смуг в високо-контрастних ділянках на зовнішніх частинах зображення. Цей ефект не варто плутати з осовою (поздовжньою) хроматичною аберациєю, яка спричиняє аберацию по всьому полю зору, зокрема фіолетовий ореол.

Дисторсія повністю відсутня у симетричних об'єктивів. Їх коефіцієнт лінійного збільшення дорівнює  $-1$ , тобто зображення перевернуто, і його розмір дорівнює розміру предмета. Це може використовуватися під час макрозйомки. Під симетрією об'єктива мається на увазі дзеркальна симетрія відносно площини апертурної діафрагми, перпендикулярної оптичній осі.

Дисторсія у об'єктивів, близьких до симетричним, зазвичай мала, навіть якщо коефіцієнт лінійного збільшення не дорівнює  $-1$ . Цей випадок найчастіше зустрічається на практиці.

У анастигматів (особливих об'єктивів), що не володіють симетрією, виправлення дисторсії можливо у зв'язку з тим, що паразитне відхилення променів при дисторсії майже не призводить до зниження роздільної здатності і набагато менш помітно, ніж порівнянне відхилення променів при інших аберациях.

Величина відносної дисторсії у нормального об'єктива зазвичай становить  $\approx 0,5\%$ . У далекофокусних об'єктивів дисторсія зазвичай менше, ніж у нормальних, а у ширококутних – більше. У деяких випадках до виправлення дисторсії пред'являються підвищені вимоги.



Іноді величина дисторсії не має значення. Об'єктиви з невиправленою дисторсією називаються дисторзуючими і застосовуються, наприклад, для метеорологічних спостережень. Дисторсія вводиться навмисно в деякі ширококутні об'єктиви для компенсації спотворень перспективи і інших недоліків.

Для розуміння теоретичної основи природи дисторсії, розглянемо деяку оптичну систему. Нехай вісь абсцис ( $x$ ) збігається з оптичною віссю системи. Площині  $\alpha$  і  $\beta$  перпендикулярні оптичній осі. Площина  $\alpha$  лежить до оптичної системи, а площину  $\beta$  – після. На площині  $\beta$  формується зображення. Промінь світла, спрямований паралельно оптичній осі, при перетині з площиною  $\alpha$  утворює точку А, проходить через оптичну систему (при цьому змінює напрямок) і при перетині з площиною  $\beta$  утворює точку В. Положення точки А представимо вектором  $\vec{r} = (y, z)$ , а точки В – аналогічним вектором  $\vec{R}$ . Вектори  $\vec{r}$  і  $\vec{R}$  лежать відповідно в площинах  $\alpha$  і  $\beta$ , починаються з точок перетину своїх площин з оптичною віссю.

Для ідеальної оптичної системи координати точки В ( $y, z$ ) будуть визначатися через координати точки А ( $y, z$ ) за такою формулою:

$$\vec{R} = b_0 \vec{r}, \quad (3.2)$$

де  $b_0$  – коефіцієнт лінійного збільшення, безрозмірна величина.

При наявності дисторсії третього порядку (а для асиметричних оптичних систем дисторсії бувають тільки непарних порядків: 3-го, 5-го, 7-го і т. п.) у формулу додають додатковий доданок:

$$\vec{R} = b_0 \vec{r} + F_3 r^2 \vec{r}, \quad (3.3)$$

де  $r$  – довжина вектору  $\vec{r}$ , м;

$F_3$  – дисторсія третього порядку (зазвичай вносить найбільший вклад в спотворення форми об'єкта),  $\text{м}^{-2}$ .

Якщо  $F_3$  має той самий знак, що і  $b_0$ , виникне «подушка», в іншому випадку – «бочка». Для дисторсії вищих порядків ( $F_n$  при  $n > 3$ ) в формулу додають по одному доданку на кожну дисторсію непарного порядку ( $F_3, F_5, F_7$  і т. д.):

$$\vec{R} = b_0\vec{r} + F_3r^2\vec{r} + F_5r^4\vec{r} + F_7r^6\vec{r} + \dots \quad (3.4)$$

При наявності дисторсії вищих порядків спотворення форми можуть мати більш складний вид, але на практиці (наприклад, у фотографії) цей випадок зустрічається рідко.

Величини  $F_n$  залежать:

- від відстані між оптичною системою і предметом, зображення якого потрібно отримати;
- від довжин хвиль світла.

Якщо потрібно враховувати вплив інших аберацій, то у вираз для  $\vec{R}$  додаються інші складові, які залежать не тільки від  $\vec{r}$ , але й від координат променю у вхідному вічку об'єктива [7].

Не слід плутати з дисторсією спотворення перспективи, викликані проектуванням тривимірного простору на площину. При таких викривлення деякі паралельні лінії на зображенні виглядають непаралельними, деякі вертикальні лінії – похилими. Але до дисторсії це не відноситься.

Дисторсія у фотографії – не завжди погано. Наприклад, в об'єктивах «fish eye» («риб'яче око») дисторсія є перевагою. Такі об'єктиви надають кут огляду до  $180^\circ$ , у результаті чого на фотографії виходить повноцінне коло з бочкоподібною дисторсією.

У комп'ютерній техніці існують доволі ефективні методи боротьби з оптичними спотвореннями на фотографіях. Дисторсія виправляється підбором лінз й інших елементів оптичної системи при її розробці. Дисторсія у цифровому зображенні може бути виправлена програмно.

### 3.2 Калібрування камери для усунення оптичних спотворень на фотографіях

У першу чергу необхідно виконати калібрування камери. Калібруванням камери називають розрахунок її внутрішніх і зовнішніх параметрів за отриманими з її допомогою фото та відео. У результаті отримується необхідна для усунення дисторсії інформація про камеру.

У розрахунках коефіцієнтів дисторсії бібліотека OpenCV враховує як тангенціальні, так і радіальні складові [8]. Для радіальної складової використовується наступна формула:

$$\begin{aligned} x_{corrected} &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{corrected} &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \end{aligned} \quad (3.5)$$

де  $r$  – відстань від точки перетину головною оптичною віссю матриці камери до точки проєкції;

$k_1, k_2$  – коефіцієнти радіального викривлення.

Таким чином, піксельна точка на спотвореному зображенні маючі координати  $(x, y)$ , отримує координати  $(x_{corrected}, y_{corrected})$  на виправленому зображенні. Присутність радіальної дисторсії виявляється в ефектах «бочки» та «риб'ячого ока» на фотографіях.

Тангенціальна дисторсія виникає через те, що лінзи об'єктива камери не можуть бути ідеально паралельні площині, яка фотографується. Її можна усунути за наступною формулою:

$$\begin{aligned} x_{corrected} &= x + [2p_1 xy + p_2(r^2 + 2x^2)] \\ y_{corrected} &= y + [p_1(r^2 + 2y^2) + 2p_2 xy], \end{aligned} \quad (3.6)$$

де  $p_1$  та  $p_2$  – коефіцієнти тангенціального викривлення.

У підсумку, можна виділити 5 параметрів спотворення, які в OpenCV представлені у вигляді однорядкової матриці з п'ятьма стовпцями:

$$Distortion_{coefficients} = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3) \quad (3.7)$$

Для перетворення координат точок реального світу в координати пікселей на фотознімку застосовується *матриця камери*:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3.8)$$

де  $f_x$  та  $f_y$  – фокусні відстані камери, а  $c_x, c_y$  – оптичні центри камери.

Присутність складової  $w$  у складі матриці піксельних координат пояснюється необхідністю дотримання відповідності розмірностей матриць координатних систем при множенні і не носить ніякого реального змісту. Невідомими параметрами є  $f_x$  і  $f_y$  – фокусні відстані камери, та  $(c_x, c_y)$ , які представляють оптичні центри, виражені в піксельних координатах. Якщо для обох осей використовується спільна фокусна відстань зі співвідношенням  $a$  (зазвичай 1), тоді  $f_y = f_x * a$  і рівняння вище буде мати єдину фокусну відстань  $f$ .

Процес знаходження матриць камери і коефіцієнтів дисторсії називається калібруванням. Розрахунки їх параметрів виконуються за допомогою геометричних рівнянь. Застосовувані рівняння залежать від типу об'єктів, що використовуються для калібрування. У даний момент OpenCV підтримує 3 типи об'єктів для калібрування:

- Класична чорно-біла шахова дошка;
- Симетричний круговий патерн;
- Асиметричний круговий патерн.

На рисунку 3.2 представлені всі 3 види калібрувальних патернів.

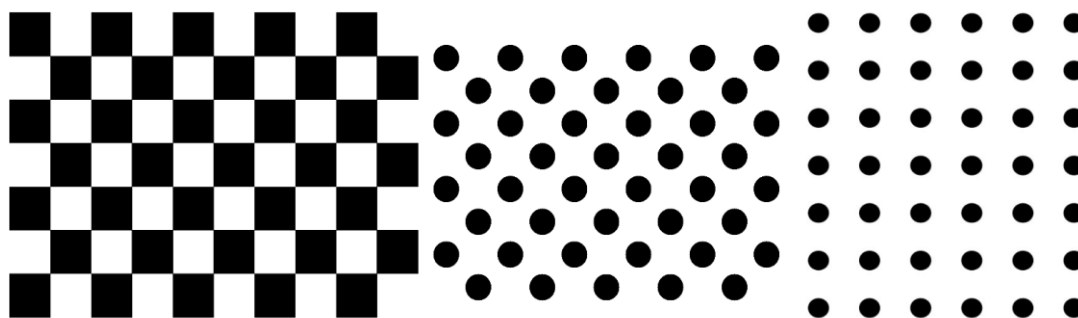


Рисунок 3.2 – Шаховий, асиметричний круговий та симетричний круговий патерни калібрування камери

Для калібрування камери необхідно зробити знімок цих патернів і знайти їх на ньому за допомогою засобів OpenCV. Кожному знайденому патерну відповідає нове рівняння. Для рішення цього рівняння треба зробити деяку кількість знімків і проаналізувати їх, щоби скласти систему з рівнянь. Наприклад, теоретично шаховий патерн передбачає мінімум 2 знімки. Але цього можна уникнути, зробивши один знімок кількох патернів одразу.

Для наших цілей краще за все підходить симетричний круговий патерн, оскільки він, по суті, потребує лише «сітку» з точок. Насправді, усі три патерни є лише обгорткою для полегшення розпізнавання. OpenCV цікавлять лише позиції вузлів патерну (точок зіткнення 4 клітинок в шаховому та позиції кругів для кругових патернів).

Таким чином, необхідно лише надрукувати і сфотографувати *калібрувальну сітку*, яка складається зі зручних для розпізнавання об'єктів, розташованих у вузлах уявної решітки з комірками у вигляді квадратів з попередньо заданою стороною. Найбільш зручними для розпізнавання об'єктами будуть вище згадані реєстраційні марки, які мають радіальну симетрію з огляду на їх круглу форму і мажуть бути легко розпізнані як камерою на різаку станка, так і засобами OpenCV. Сторона квадрата особливого значення для розпізнавання не має, але мусить бути достатньо великою, щоби вузли сітки зливались між собою на фотографії, і в той же час достатньо малою, щоби забезпечити велику точ-

ність позиціонування. Останнє твердження витікає з того, що калібрувальна сітка в нашому випадку служитиме для двох цілей – як патерн для калібрування камери і для утворення основи системи позиціонування об'єктів на різальній поверхні, адже ми попередньо знаємо позиції точок кругового патерну одна відносно одної.

### 3.3 Створення системи позиціонування на основі калібрувальної сітки

Розроблювана СКЗ передбачає нерухоме закріплення камери над різальною поверхнею, з чого слідує, що один і той самий піксель на різних фотографіях камери буде представляти одну й ту саму точку в просторі (на різальній поверхні). Залишається лише задати точку відліку і встановити опорні точки на різальній поверхні з відомими координатами, відносно яких будуть розраховуватись координати всіх інших.

Позиціонування в межах самої калібрувальної сітки є найпростішою задачею, оскільки крок сітки (сторона квадрата комірки) відомий заздалегідь. Зробивши знімок калібрувальної сітки (*калібрувальний знімок*), достатньо тільки розрахувати відношення відстані в міліметрах до відстані в пікселях між вузлами сітки і користуватись його значенням для знаходження відстаней в межах сітки. Наприклад: між вузлами калібрувальної сітки реальна відстань складає 50 мм, а на знімку – 100 пікселів. Співвідношення мм на 1 піксель (mm/pх) складає  $\frac{1}{2}$ . Знаючи, що відстань на фотографії між об'єктами складає, скажімо, 482 пікселі, достатньо помножити це значення на розраховане раніше відношення mm/pх, щоб дізнатися реальну відстань в міліметрах: 241 мм.

Для позиціонування відносно точки відліку різача необхідно разове застосування локальної камери, котра повинна знайти кожен вузол калібрувальної сітки і надати його координати, знайдені з високою точністю, системі комп'ютерного зору, яка «прив'яже» реальні координати вузлів в міліметрах відносно точки відліку до їх координат на фотографіях в пікселях. Це буде єдиний етап роботи розроблюваної системи, коли треба застосовувати камеру різача для пошуку марок. Заново виконувати таку процедуру потрібно буде лише у випадку необхідності зміни калібрувальної сітки (з іншим кроком, патерном тощо) і повторного виконання калібрування вже для нової сітки.

Подальша схема позиціонування нічим не відрізняється від поданої вище. Проте точність буде вищою, оскільки без даних від локальної камери ми можемо знати лише заданий при виготовленні крок між вузлами калібрувальної сітки, який в процесі її виготовлення (при друкуванні) може бути спотворений в силу різних технічних факторів.

Окремо слід зазначити про можливість точного позиціонування в умовах значної товщини матеріалу об'єкта. Тут проблема полягає у тому, що калібрувальна сітка зазвичай наноситься на широкий лист паперу та має майже нульовий рівень відносно робочої поверхні плоттера. А як щодо матеріалів з товщиною більше ніж 1 сантиметр? На фотографії позиції реєстраційних марок будуть мати деяке оптичне зміщення відносно координат калібрувальної сітки. Іншими словами, ті самі піксельні координати для матеріалів різної товщини будуть мати різні координати в дійсності. Геометричний зміст проблеми позиціонування марок матеріалів з високою товщиною проілюстровано на рисунку 3.3.

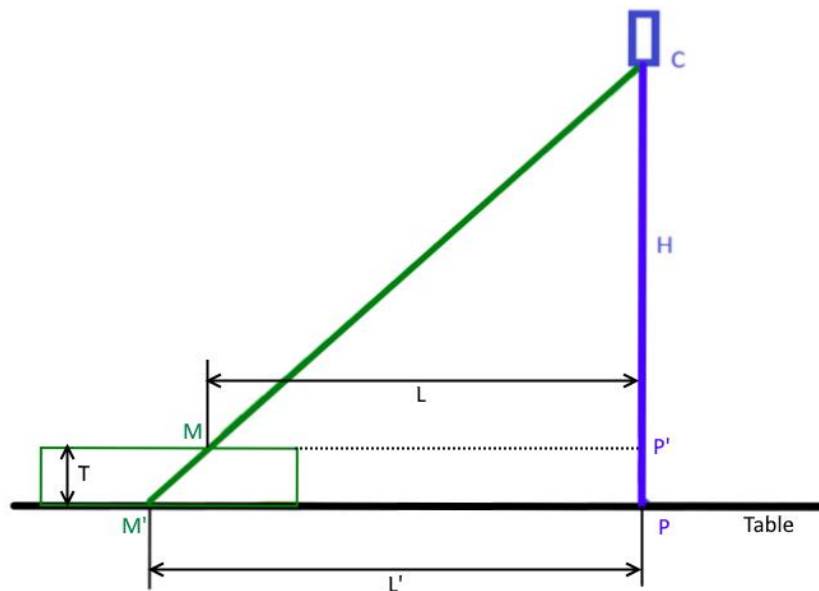


Рисунок 3.3 – Оптичне накладання точок на різному рівні відносно різальної поверхні

На рисунку 3.3 маємо:

- Точку проєкції камери на столі  $P$  – точку, у якій перпендикуляр від камери падає на стіл. Для зручності ми припускаємо, що оптична вісь камери збігається з перпендикуляром столу. Оскільки камера має бути розташована під прямим кутом до робочої поверхні плоттера, координати  $P$  в пікселях співпадають з центром знімку. Камера розташована на висоті  $H$  над рівнем різальної поверхні плоттера.
- Промаркований об'єкт для нарізки з ненульовою товщиною  $T$ , розташований на різальній поверхні.
- Центр деякої марки об'єкта  $M$ , який видно з позиції камери під кутом до площини фотографії як точку  $M'$ . Саме для точки  $M$  ми повинні розрахувати її точне положення на столі, враховуючи її висоту над поверхнею станка.

Оскільки точка  $P$  – єдина точка, видима під кутом  $90^0$  до площини поверхні плоттера, тільки її положення на фотографії буде збігатись з її дійсним положенням. Тому ми можемо використовувати її як опорну точку для розрахунку положення усіх інших точок на фотографії. Для знаходження скорегованого положення довільної точки  $M$  нам спочатку потрібно знати дійсну відстань  $L$  від її ортогональної проєкції на площину поверхні плоттера до точки  $P$ . Для цього достатньо знати висоту над рівнем робочої поверхні  $T$ , висоту камери над робочою поверхнею  $H$  та видиму відстань  $L'$  від точки  $M'$  до точки падіння перпендикуляру від камери на робочу поверхню. Далі можна застосувати геометричні закони, які свідчать про пропорційність сторін подібних трикутників.

Трикутники  $CPM'$  та  $CP'M$  подібні, оскільки є прямокутними та мають спільний гострий кут. Тоді для них справедливе наступне відношення:

$$\frac{L}{L'} = \frac{H-T}{H} \quad (3.8)$$

де  $H - T$  і  $L$  – сторони  $CP'$  та  $PM'$  трикутника  $CPM'$ ,

$H$  і  $L'$  – сторони  $CP$  та  $P'M$  трикутника  $CP'M$ .



З формули (3.8) виходить що:

$$L = L' \times \frac{H-T}{H} \quad (3.9)$$

Значення  $H$  буде змінюватись в залежності від параметрів монтажу камери та встановлюватись в якості параметра системи. Товщину матеріалу  $T$  беремо з шаблону об'єкта розпізнавання, а  $L'$  розраховуємо як відстань між двома точками на робочій поверхні плоттера через дані отримані під час калібрування. Так, з формули (3.9) ми отримуємо дійсну відстань на різальній поверхні між точками падіння на неї перпендикулярів від камери та марки на поверхні об'єкта нарізки. Знаючи координати точки  $P$  та відстань  $L$  можна визначити реальні координати точки  $M$ , зміщуючись уздовж вектору  $\overrightarrow{PM'}$  від  $P$  на  $L$  міліметрів.

Для демонстрації важливості врахування товщини матеріалу наведемо невеликий приклад. Нехай камера знаходиться на висоті 2100 мм над різальною поверхнею плоттера, товщина матеріалу становить всього 10 мм, а марку, для якої виконується позиціонування, видно на відстані 1000 мм від точки падіння перпендикуляру від камери до різальної поверхні. За формулою (3.9), дійсна відстань від марки до перпендикуляру становитиме  $\frac{1000 \cdot 2090}{2100} = 995.2$  мм. Як бачимо, нехтування товщиною спричинило зсув розрізу на майже 5 мм, що могло б призвести до псування цінного матеріалу та грошових втрат.

Усе сказане вище свідчить про важливість міцної фіксації камери відносно різальної поверхні. Найменше її зміщення призведе до невідповідності наступних знімків калібрувальному та необхідності виконувати процес калібрування заново, який, враховуючи час на пошук вузлів камерою різача і в залежності від розмірів стола та калібрувальної сітки, може зайняти досить багато часу.

У контексті описуваної системи комп'ютерного зору, процес калібрування включає не тільки знаходження матриць властивостей камери. Це також процедура повної

підготовки системи до подальшої коректної роботи з ідентифікації та позиціонуванню об'єктів на різальній поверхні на основі калібрувальної фотографії та даних від камери різача.

### 3.4 Реалізація процедури калібрування системи комп'ютерного зору

Після отримання даних від обох камер, інформації про розміри та крок калібрувальної сітки та команди запуску калібрування, програма починає пошук реєстр. марок на калібрувальній фотографії. Процедуру пошуку марок буде детально описано в наступному розділі даної дисертації. Результати пошуку у вигляді списку координат пікселів центрів марок на фотографії проходять процедуру пошуку закономірностей розташування, на основі яких будуються об'єкти калібрувальних сіток. Основу цих об'єктів складають три двомірні масиви точок, розміри яких співпадають з розмірами сіток, а комірки призначені для збереження координат вузлів сітки, тобто центрів марок. Перший масив зберігає координати вузлів сітки в піксельних координатах фотографії, а другий – отримані від камери різача відповідні координати реального світу в міліметрах. Третій призначений для зберігання неспотворених координат вузлів на «вирівняній» фотографії.

Після побудови всіх об'єктів сіток відповідно до кількості реальних сіток на фотографії та заповнення їх перших масивів в порядку реального розташування вузлів, відбувається пошук відповідності розташування комірок першого масиву та точок, отриманих від камери різача. Таким чином, встановлюється однозначна відповідність реальних координат вузлів калібрувальних сіток в міліметрах до їх координат на фотографії в пікселях та заповнюються другі масиви об'єктів сіток. На основі отриманих даних визначається також позиція точки відліку (початку координат) на фотографії.

Далі слідує власне калібрування камери. Для отримання калібрувальних матриць OpenCV пропонує метод `calibrateCamera()`, який розраховує внутрішні та зовнішні коефіцієнти камери через надану йому інформацію про позиції калібрувальних патернів на

знімках з різних (або одного) ракурсів. Він приймає координати точок патернів (калібрувальних сіток) у просторі, їх координати на фотографії (перші два масиви об'єктів сіток), а також пусті матриці для власних коефіцієнтів та коефіцієнтів спотворення камери, які заповнює розрахованими значеннями. Вхідний вектор містить стільки елементів, скільки було зроблено знімків патернів. Якщо той самий калібрувальний патерн сфотографовано кілька разів з одного ракурсу, і він повністю видимий, усі вектори будуть однаковими. Також можна використовувати неповні шаблони або навіть різні моделі в різних ракурсах. Тоді вектори будуть різними. Точки є тривимірними, але оскільки вони знаходяться в системі координат патерну, то, якщо платформа є плоскою, можна перевести модель на двовимірну систему координат, так що Z-координата для кожної точки вхідного об'єкта дорівнюватиме 0.

Принцип роботи `calibrateCamera()` базується на «Новій гнучкій техніці калібрування камери» [], описаній Женгйюу Зангом у 2000 році для шахових патернів, а також алгоритмі Жан-Іва Буге, розробленого для системи Matlab. Координати точок 3D-об'єкта (патерна) та їх відповідні 2D-проекції в кожному ракурсі повинні бути обов'язково вказані. Це досягається завдяки заздалегідь відомій геометрії об'єкта та легкому виявленні його опорних точок за допомогою функціоналу OpenCV. OpenCV має вбудовану підтримку розпізнавання шахової дошки та її вузлів для калібрувальних цілей (функція `FindChessboardCorners()`).

Алгоритм роботи функції `calibrateCamera()` передбачає виконання наступних кроків:

- 1) Обчислення початкових внутрішніх параметрів камери (виконується лише для плоских моделей калібрування) або їх зчитування із вхідних параметрів. Усі коефіцієнти спотворень спочатку встановлюються в нулі, за винятком тих, які були встановлені прапорцями `CV_CALIB_FIX_Kx` в якості вхідних параметрів функції.
- 2) За допомогою функції `solvePnP()` оцінюється початкове положення камери, нібито її внутрішні параметри вже відомі.

3) Запускається алгоритм Левенберга – Марквардта для оптимізації та зведення до мінімуму похибку відтворення, тобто загальну суму квадратних відстаней між видимими точками патерну (координати в пікселях) та спроектованими на модель матриці камери (за допомогою поточних оцінок для параметрів камери та її позиції) координатами тих самих точок в реальному світі. Для цього використовується функція `OpenCV projectPoints()`.

Отримані матриці внутрішніх параметрів камери та коефіцієнтів спотворення зберігаються і далі використовуються для розрахунку «правильних» координат вузлів калібрувальної сітки за допомогою функції `OpenCV undistortPoints()`. Ця функція розраховує скореговані координати точок (пікселів) на зображення, отримуючи на вхід їх початкові координати, матрицю внутрішніх параметрів камери та вектор коефіцієнтів спотворень. Отримані координати відповідають тим, які мають ті самі об'єкти на «вирівняному» зображенні без впливу дисторсії.

Принцип роботи процедури `undistortPoints()` подібний до функцій `undistort()` та `initUndistortRectifyMap()`, за винятком того, що вона працює з довільним набором точок, а не з цілим зображенням. Також функція виконує зворотну трансформацію до тої, що виконується в `projectPoints()`. У випадку тривимірного об'єкту, реконструкції його 3D-координат не відбудеться, оскільки метод працює лише для площин, розраховуючи навіть вектор зсуву, якщо функції було передано правильний параметр  $R$  – коригуюча матриця (3x3) для простору об'єкта. Якщо цей параметр пустий – ніякого перетворення не відбувається.

Якщо ж виникає необхідність отримати повністю позбавлене впливу дисторсії зображення, можна скористатись функцією `OpenCV undistort()`, яка, приймаючи на вхід спотворене зображення, матрицю внутрішніх параметрів камери та вектор коефіцієнтів спотворень на виході, створює нове вирівняне зображення, вільне від впливу радіальних і тангенціальних оптичних спотворень. Пікселі вихідного зображення, для яких немає відповідних пікселів на вхідному, заповнюються нулями, тобто залишаються чорними.

На рисунку 3.4 зображено фотографія клавіатури з сильним ефектом «бочки» та її вирівняна за допомогою функції `undistort()` копія.



Рисунок 3.4 – Викривлене оптично, та вирівняне методами OpenCV зображення клавіатури

Перераховані неспотворені координати записуються в третій масив об'єктів сіток. У подальшому всі точки, знайдені на фотографіях, будуть проходити таку обробку для підвищення точності позиціонування.

На цьому процес калібрування закінчується. У результаті програма комп'ютерного зору отримує всю необхідну інформацію для точної ідентифікації та позиціонування об'єктів на наступних фотографіях різальної поверхні.

## 4 ОБРОБКА ЗОБРАЖЕННЯ ТА ПОШУК РЕЄСТРАЦІЙНИХ МАРОК

### 4.1 Засоби OpenCV для пошуку реєстраційних марок

Найголовнішою задачею розроблюваної системи комп'ютерного зору є пошук всіх реєстраційних марок на фотографії. На щастя, для цього бібліотека OpenCV пропонує широкий набір інструментів для пошуку різноманітних геометричних об'єктів на растрових зображеннях. Трохи спрощує задачу сама форма марки – круг переважно чорного кольору. На бінарному зображенні це виглядатиме як скупчення темних пікселів з округлим контуром. Спеціально з ціллю пошуку таких об'єктів було розроблено клас SimpleBlobDetector [9]. Цей клас реалізує простий алгоритм для знаходження чорних «плям», до яких як раз відносяться реєстр. марки на зображенні.

Спочатку виконується конвертація кольорового зображення в кілька чорно-білих з застосуванням різних порогів від мінімально до максимально заданих (або прийнятих за замовчуванням) з деяким кроком. Суть застосування порогів полягає в порівнянні значення пікселю зі значенням порогу. Попередньо зображення з кольорового переводиться у відтінки сірого. Далі, якщо значення пікселю більше за значення порогу, йому присвоюється один колір (наприклад чорний), а якщо менше – інший (наприклад білий). Далі алгоритм проводить пошук контурів на отриманих чорно-білих зображеннях та знаходить їх центри. Потім знайдені на різних зображеннях контури групуються по координатам. Близько розташовані центри формують єдину групу, котра відповідає окремій «плямі». І, врешті решт, аналізуючи кожну групу, алгоритм знаходить остаточні центри та радіуси «плям» та повертає їх у формі колекції структур, що представляють координати та властивості знайдених «плям».

Також клас перед поверненням результатів піддає фільтрації за заданими параметрами знайдені «плями». До цих параметрів відносяться:

1. Яскравість. Фільтр порівнює інтенсивність кольору в центрі «плями» з заданим параметром `blobColor`. Якщо вони сильно різняться, «пляма» відкидається.

2. Площа. Допускаються лише «плями», що мають площу від заданої параметром  $minArea$  (включно) до  $maxArea$  (не включно). Тобто встановлення  $minArea$  рівним 100, відкине всі контури з площею менше 100 пікселей.
3. Округлість. Тут оцінюється наскільки сильно «пляма» схожа на круг. Наприклад шестикутник буде більш округлим, аніж, скажімо, квадрат. Треба лише встановити відповідне значення параметрам  $minCircularity$  та  $maxCircularity$  (від 0 до 1). Округлість розраховується за формулою:

$$\frac{4\pi Area}{(perimeter)^2} \quad (4.1)$$

де  $Area$  – площа плями що розглядається, а  $perimeter$  – її периметр.

З формули 4.1 слідує, що ідеальний круг має округлість рівну 1, квадрат – 0,785 і так далі.

4. Опуклість. Ступінь опуклості визначається як відношення площі контуру до площі її опуклої оболонки. Опуклою оболонкою контуру є найменший опуклий контур, який повністю включає в себе заданий. Для включення цього фільтра необхідно встановити значення  $minConvexity$  та  $maxConvexity$  (обидва від 0 до 1).

5. Коефіцієнт інерції. Вимірюється ступінь видовженості контуру. У круга цей коефіцієнт рівний 1, у еліпса від 0 до 1, а в лінії рівно 0. Для використання фільтрації за коефіцієнтом інерції необхідно встановити параметри  $minInertiaRatio$  та  $maxInertiaRatio$  відповідно.

Вплив перелічених вище параметрів на якість пошуку та критерії відбору «плям» на зображенні продемонстровано на рисунку 4.1.

Однак у класу `SimpleBlobDetector` є один суттєвий недолік – його ефективність сильно залежить від якості зображення, адже на фотографії реєстраційна марка буде мати вигляд скоріше плями з градієнтним переходом від більш темного центру до світліших крайніх точок з нерівними контурами. Тому `SimpleBlobDetector` буде використо-

уватись у розроблюваній системі комп'ютерного зору лише для цільового пошуку марок у заданому регіоні, де заздалегідь відомо що марка там є, і його можна використати з нижчими вимогами до правильності форми шуканих об'єктів.

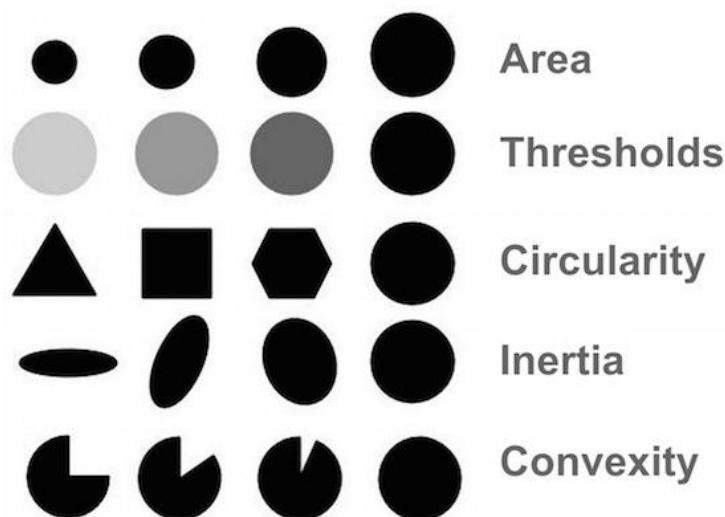


Рисунок 4.1 – Вплив параметрів SimpleBlobDetector на критерії відбору шуканих об'єктів

Що стосується інших варіантів пошуку реєстраційних марок, то тут варто знову звернути увагу на геометричні властивості марки. Як не повертай круг однорідного кольору навколо його центру – його вигляд завжди залишатиметься однаковим. Така його властивість дозволяє застосувати до нього метод *співставлення зразка*.

Співставлення зразка — це метод пошуку положення зразка на зображенні більшого розміру. Іншими словами, маючи шматочок зображення, можна знайти його положення на цілому зображенні. В OpenCV цей метод реалізує функція `matchTemplate()`. Вона просто переміщує зразок по цільовому зображенню та порівнює зразок з поточним шматочком під ним. Існує декілька реалізацій методу співставлення зразка в OpenCV. Вибрати конкретний можна встановленням значення одного з параметрів `matchTemplate()`. Повертає функція сіре зображення, де кожен піксель показує наскільки сусідня з ним область співпадає зі зразком [10].



Якщо основне зображення має розміри ( $W \times H$ ), а зразок ( $w \times h$ ), вихідне зображення матиме розміри ( $W - w + 1, H - h + 1$ ). Як тільки результуюче зображення отримано, можна використовувати функцію `minMaxLoc()` для отримання координат мінімального (максимального – у залежності від використаного методу співставлення зразка), за шкалою сірого пікселя. Це буде верхній лівий кут області, що відповідає зразку.

Однак такий метод підходить лише для випадку, коли розшукується єдине співпадіння заданого зразка на цільовій фотографії. Якщо таких співпадінь очікується більше одного, необхідно використовувати поріг відтінків сірого [11]. Якщо метод передбачує мінімальне значення пікселю (світлий) в області співпадіння, то необхідно встановити поріг від 0 до 1 ближче до 0 і циклічно викликати функцію `minMaxLoc()`, порівнювати найменше значення, яке вона повертає із заданим порогом. Якщо воно менше, то можна зберігати його координати та зафарбовувати вже знайдену область в колір зворотній шуканому (для мінімуму – чорний) для запобігання повторного повернення тієї ж самої координати. Якщо ж `minMaxLoc()` повернула мінімальне значення більше порогового, значить можна переривати цикл – достатньо чіткі співпадіння закінчились.

Для методів, які повертають співпадіння з максимальним значенням відтінку сірого (темним) пікселя, загальний принцип зберігається, але поріг береться ближче до 1, а приймаються точки з більшим, ніж порогове, значенням відтінку.

Усього алгоритм співставлення зразка пропонує 4 методи порівняння пікселів зразка з пікселями цільового зображення, кожен з яких відрізняється формулою для розрахунку  $R(x, y)$  – значення пікселю з координатами  $(x, y)$  вихідного зображення, результату порівняння пікселя  $(x', y')$  зразка  $T$  з пікселем вхідного зображення  $I(x + x', y + y')$ . При чому  $x' = 0 \dots w - 1, y' = 0 \dots h - 1$ , де  $w$  і  $h$  – ширина та висота зразка у пікселях.

- Метод `CV_TM_SQDIFF`:

$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2 \quad (4.2)$$

- Метод CV\_TM\_SQDIFF\_NORMED:

$$R(x, y) = \frac{\sum_{x',y'} (T(x',y') - I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}} \quad (4.3)$$

- Метод CV\_TM\_CCORR:

$$R(x, y) = \sum_{x',y'} (T(x',y') \cdot I(x+x',y+y')) \quad (4.4)$$

- Метод CV\_TM\_CCORR\_NORMED:

$$R(x, y) = \frac{\sum_{x',y'} (T(x',y') \cdot I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}} \quad (4.5)$$

- Метод CV\_TM\_CCOEFF:

$$R(x, y) = \sum_{x',y'} (T'(x',y') \cdot (I'(x+x',y+y'))) \quad (4.6)$$

- Метод CV\_TM\_CCOEFF\_NORMED:

$$R(x, y) = \frac{\sum_{x',y'} (T'(x',y') \cdot I'(x+x',y+y'))^2}{\sqrt{\sum_{x',y'} T'(x',y')^2 \cdot \sum_{x',y'} I'(x+x',y+y')^2}} \quad (4.7)$$

де  $T'(x',y') = T(x',y') - \frac{1}{w \cdot h} \cdot \sum_{x'',y''} T(x'',y'')$ ,

$I'(x+x',y+y') = I(x+x',y+y') - \frac{1}{w \cdot h} \cdot \sum_{x'',y''} I(x+x'',y+y'')$

На рисунку 4.2 показано результати пошуку на деякому зображенні способом співставлення зразка з використанням методу CV\_TM\_CCOEFF\_NORMED, вирізаного з нього шматочку.



Рисунок 4.2 – результат пошуку зразка на фотографії методом  $CV\_TM\_CCOEFF\_NORMED$

Використання співставлення зразка в описуваній системі потребує лише створення кількох зразків марок різного кольору, діаметру, освітленості та інших параметрів, що впливають на її вигляд. Далі можна застосовувати пошук відповідного зразка в залежності від того, які марки представлені на фотографії. Створити шаблон марки можна вручну, її вирізанням з цільового зображення за допомогою фоторедакторів. Але такий спосіб дуже незручний, бо потребує зайвих витрат часу. Тому для демонстраційної програми було розроблено інструмент пошуку марки автоматично через графічний інтерфейс, принцип роботи якого базується на використанні SimpleBlobDetector для деякої невеликої області зображення, вказаної користувачем, з невисокими вимогами до правильності форми шуканої марки.

#### 4.2 Підготовка зображення, та пошук реєстраційних марок на ньому

У загальному, процес пошуку марок на зображенні можна розділити на наступні етапи:

- 1) Усунення впливу оптичних дисторсій на зображення, якщо це вже можливо.
- 2) Розділення цільового зображення на дрібніші області.
- 3) Пошук марок методом співставлення зразка на кожній області зображення у паралельному режимі.

#### 4) Комбінування результатів та усунення дублікатів.

Пошук марок на зображенні може відбуватись як після, так і перед калібруванням камери, тому вирівнювання зображення функцією `undistort()` відбувається лише для пошуку марок перед ідентифікацією об'єктів для нарізки, коли калібрування вже обов'язково відбулося і є змога виправити оптичні спотворення. Коли програма виконує задачу пошуку марок на фотографії симетричного кругового патерну для потреб калібрування—позиціонування, вирівнювання зображення відбувається вже після знаходження всіх марок, як основи для розрахунку матриці параметрів камери та коефіцієнтів дисторсії.

Сучасні багатопоточні процесори дозволяють паралельне виконання різних обчислювальних операцій, суттєво скорочуючи загальний час обчислень. Для задачі пошуку марок за зразком це досягається шляхом ділення зображення на дрібніші ділянки, які далі оброблюються паралельно. Вхідне зображення розділяється на квадрати зі стороною  $250 + n$  пікселів, де  $n$  – додаткова кількість пікселів рівна розміру найбільшого наявного шаблону реєстраційної марки (приблизно дорівнює діаметру найбільшої марки).

Додавання числа пікселів достатнього, щоб вмістити найбільшу марку, зумовлене необхідністю забезпечити перекриття суміжних зон між областями зображення, де теоретично можуть знаходитись марки. Якщо цього не робити, то марки, які опинились на зрізі, будуть розділені між зображеннями, і в цілому вигляді не потраплять на жодну область, а значить не будуть розпізнані. Графічне зображення такої ситуації подано на рисунку 4.3.

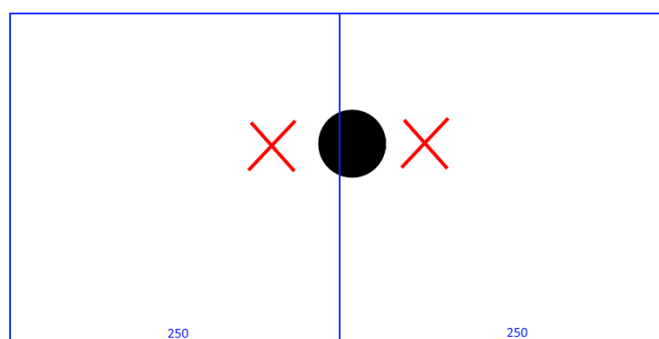


Рисунок 4.3 – Розділення марки при поділі основного зображення на менші, без перекриваючих зон

Якщо ж при поділі вхідної фотографії враховувати подібні ситуації та використовувати перекриття сусідніх областей зоною достатнього розміру, така ситуація стає в принципі неможливою. Якщо марка потрапляє в одну область лише частково, це означає, що в іншу область вона потрапить повністю. Така ситуація зображена на рисунку 4.4.

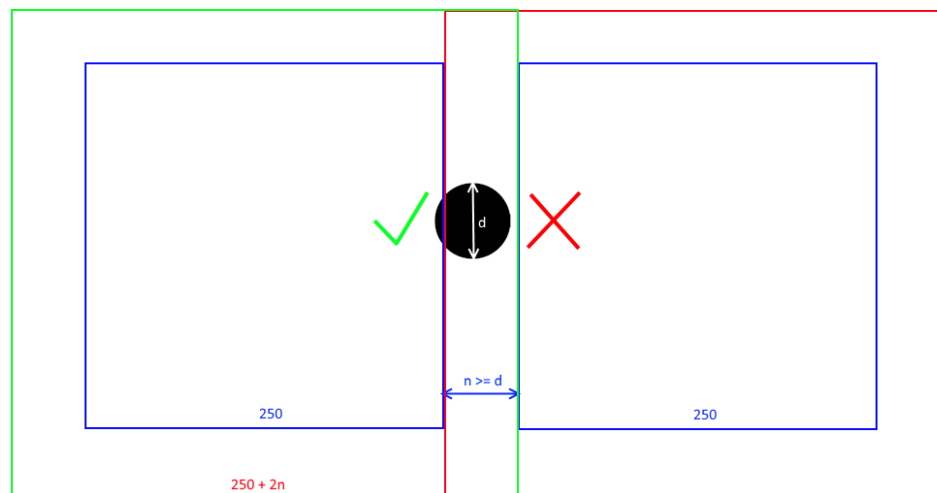


Рисунок 4.4 – Захоплення марки перекриваючою зоною при поділі основного зображення

При діленні початкового зображення на менші, для кожної області запам'ятовується її зсув відносно лівого верхнього кута вихідної фотографії. Після завершення пошуку марок на кожній області, маємо набір координат знайдених марок відносно точки відліку (верхнього лівого кута) кожної області. Тепер під час об'єднання всіх результатів пошуку, до кожної знайденої марки додаємо зсув її області по відношенню до основного зображення і отримуємо координати марки в пікселях на початковому зображенні робочої поверхні плоттера.

Окрім позитивного ефекту унеможливлення втрати марок при діленні на області, їх перекриття також вносить деяку вірогідність дублювання результатів пошуку при потраплянні марки повністю на зону перекриття. Також утворення дублікатів можливе за рахунок використання кількох різних зразків марок для методу співставлення зразка. Для пошуку та видалення дублікатів у подвійному циклі координати кожної марки по-

рівнюються з іншими. Якщо вони знаходяться менше, ніж на відстані радіусу найменшого марку (береться найменший розмір шаблону марку), то одна з них видаляється як дублікат.

Тільки після усунення дублікатів, марку можна передавати далі для калібрування або ідентифікації об'єктів для нарізки серед них.

Єдиною проблемою такого методу пошуку є можливість потрапляння координат зайвих об'єктів до списку знайдених марок. Частіше за все об'єкти для нарізки мають на собі якесь зображення, яке може містити елементи схожі на чорні кола, і будуть також розпізнані на етапі пошуку реєстраційних марок. Одним з можливих рішень може бути зменшення порогу допуску для співставлення зразка. Однак, так скоріш за все будуть відкинуті і потрібні, але невдало сфотографовані марку з тих чи інших причин (освітлення, тіні і т. п.). Краще за все буде відкинути зайві марку вже на етапі пошуку об'єктів нарізки.

## 5 ІДЕНТИФІКАЦІЯ ОБ'ЄКТІВ НАРІЗКИ

### 5.1 Алгоритми узгодження множин точок

Алгоритм пошуку відповідностей шаблонам нарізки полягає у використанні деяких елементів з теорії графів. Марки шаблону мають координати та утворюють вершини зваженого графа з ребрами, вага яких дорівнює відстані між марками шаблону. Залишається лише знайти такий самий граф серед усіх знайдених на знімку реєстр. марок. Зрозуміло, що пошук марок допускає, що деякі марки на фото можуть бути не знайдені через недосконалість алгоритмів пошуку, якість зображення, освітлення тощо. Це означає, що відсутність деяких вершин цільового графа серед знайдених на знімку реєстр. марок є допустимою, якщо з урахуванням всіх інших наявних марок, можна встановити однозначну їх відповідність шуканому шаблону об'єкта нарізки.

Оскільки координати марок шаблону дані в міліметрах відносно невідомого нам початку координат матеріалу, на який ці марки нанесені, оперувати можна лише абсолютними значеннями відстаней між ними. Результатом процедури калібрування, яка має бути обов'язково виконана перед ідентифікацією об'єктів нарізки, є система позиціонування, яка дозволяє досить точно перевести піксельні відстані між точками на фотографії у відповідні відстані на різальній поверхні у міліметрах. Так, для кожної знайденої марки ми можемо знайти всі відстані від неї до інших (ребра графа) та порівняти їх з набором ребер між марками з шаблону, виявивши співпадіння. У цьому і полягає основна ідея алгоритму пошуку відповідностей набору знайдених реєстр. марок заданим шаблонам об'єктів нарізки. Принципово розроблювана СКЗ вирішує задачу узгодження множин точок.

Узгодження множин точок (англ. point set registration, англ. point matching) у теорії розпізнавання образів та комп'ютерному зорі є процесом знаходження просторового перетворення, яке узгоджує дві множини точок. Метою знаходження такого перетворення є злиття декількох множин точок у єдину модель і відображення нового вимірювання на відомий набір даних для виявлення ознак або оцінювання його положення. Множина

точок може бути вхідними даними з 3D-сканера або масивом, отриманим від далекоміра. Для використання в обробці зображень і реєстрації зображень на основі ознак, множина точок може бути множиною ознак, отриманою виділенням ознак із зображення, наприклад, виявленням кутів. Узгодження множин точок використовується при оптичному розпізнаванні символів, у доповненій реальності та суміщенні даних магнітно-резонансної томографії зі знімками комп'ютерної томографії. Проблема узгодження множин точок проілюстрована на рисунку 5.1.

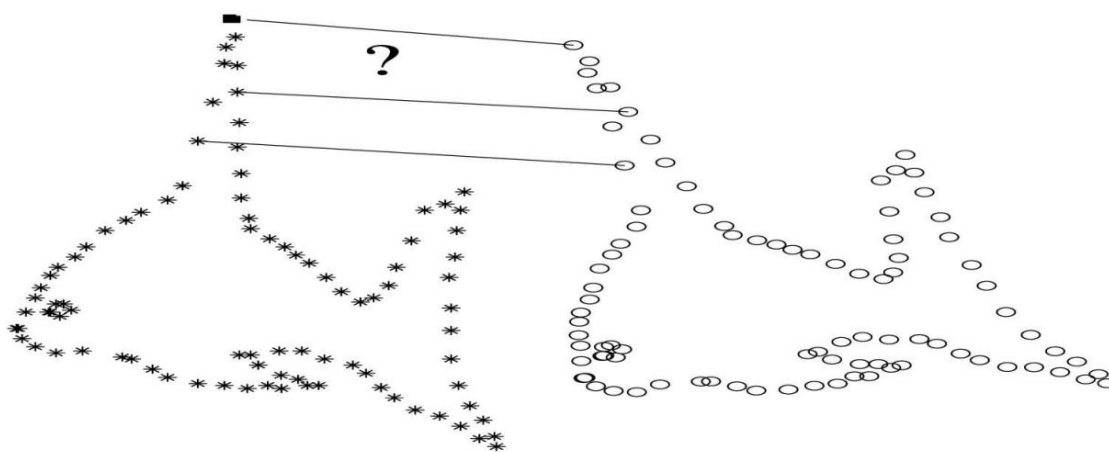


Рисунок 5.1 – Ілюстрація задачі узгодження двох множин точок

Усі методи узгодження множин точок можна поділити на два принципово різні класи: *жорсткого* узгодження і *гнучкого* узгодження множин точок.

Маючи два набори точок, жорстке узгодження дає жорстке перетворення, яке відображає одну множину точок на іншу. Жорстке перетворення визначається як перетворення, що не змінює відстані між будь-якими двома точками. Як правило, таке перетворення складається з переносу та обертання. У рідкісних випадках набір точок також може бути відображеним.

Гнучкі перетворення включають афінні перетворення, такі як масштабування та зсувне відображення. Проте у контексті узгодження множин точок, гнучке узгодження зазвичай передбачає нелінійну трансформацію. Якщо відомі власні нормальні коливання



набору точок, нелінійне перетворення може бути параметризоване власними значеннями.

Деякі підходи до узгодження множин точок використовують алгоритми, які вирішують більш загальну задачу співставлення графів. Проте обчислювальна складність таких методів, як правило, висока, і вони обмежені жорсткими перетвореннями. Нині існує декілька відомих алгоритмів, специфічних для задачі узгодження множин точок. Найпростіший та найпопулярніший серед них – ітеративний алгоритм найближчих точок.

Ітеративний алгоритм найближчих точок (англ. Iterative Closest Point - ICP) – алгоритм, що використовується для зведення до мінімуму різниці між двома множинами точок. ICP часто використовується для відновлення двовимірних (2D) або тривимірних (3D) поверхонь із різних сканів для визначення місця розташування роботів і планування оптимального їх шляху (особливо коли одометрія коліс ненадійна внаслідок слизького ландшафту), реєстрації моделі кістки тощо [12].

Алгоритм концептуально простий і часто використовується в режимі реального часу. Він багатократно застосовує перетворення (зміщення, обертання), необхідні для доведення до мінімуму відстані між точками з двох наборів.

Ще один алгоритм, запропонований Стівеном Голдом у 1998 році, називається «прямим узгодженням точок» (англ. Robust point matching – RPM) [13]. Цей алгоритм виконує узгодження за допомогою методу імітації відпалу – загального алгоритмічного методу розв’язання задачі глобальної оптимізації, у якому процедура пошуку глобального розв’язку імітує фізичний процес відпалу та гнучкого підбору відповідностей між наборами точок. У той час як в ICP відповідність, формована евристикою найближчого сусіда, є бінарною, RPM використовує гнучке співставлення, коли відповідність між будь-якими двома точками може лежати між 0 та 1, хоча у підсумку зводиться до 0 або 1.

Наступний алгоритм називається кореляцією ядра (англ. Kernel correlation – KC) і у порівнянні з двома попередніми є більш жорстким до спотворених даних (точки мають

зміщення, так що не можна встановити повної відповідності) [14]. На відміну від ICP, де кожна точка одної множини розглядається в контексті встановлення відповідності лише одній точці з іншої множини, у КС враховуються одночасно всі точки. Таким чином, це багатозв'язний алгоритм узгодження.

І, нарешті, обраний у якості допоміжного алгоритму у розроблюваній системі комп'ютерного зору, метод носить назву «Узгоджений дрейф точок» і є досить новим у порівнянні з попередньо описаними алгоритмами узгодження наборів точок [15]. Узгоджений дрейф точок (англ. Coherent point drift (CPD)) був представлений у 2006 році сумісно українцем Андрієм Мироненко та китайкою Ксубо Сонг. Алгоритм застосовує параболічний підхід до узгодження множин точок, подібно до модифікованого методу кореляції ядра з використанням моделі змішування Гаусса. У той час як ICP мінімізує відстані між точками, CPD використовує моделі змішування Гаусса для мінімізації похибки положення точки відносно всіх інших точок.

Алгоритм узгодженого дрейфу точок має реалізацію у відкритому доступі та показує високу ефективність і точність. Єдиним його недоліком є значне сповільнення при збільшенні розмірності вхідних даних у зв'язку з високою складністю і кількістю виконуваних у ньому обчислень. Але навіть при цьому такий метод значно швидший, ніж пошук марок за допомогою інтегрованої в ріжучий механізм камери. Порівняльні дослідження точності та швидкодії показали значну перевагу CPD над його найближчими конкурентами ICP та RPM [15]. Тому саме цей алгоритм обрано в якості механізму остаточного формування результату розпізнавання в розроблюваній системі.

## 5.2 Процедура ідентифікації об'єктів цифрової нарізки

Ідентифікація об'єктів нарізки починається зі створення знімку різальної поверхні глобальною камерою. Разом зі знімком системі комп'ютерного зору також на вхід подається шаблон(и) об'єктів розпізнавання, які присутні на знімку, та зразки марок для розпізнавання, якщо планується використовувати метод співставлення зразка. Після отримання всіх даних та відповідної команди СКЗ починає процес ідентифікації.

Спочатку відбувається пошук реєстраційних марок описаним у попередньому розділі способом. Результат пошуку у вигляді списку координат центрів знайдених реєстр. марок подається на вхід процедури пошуку відповідностей шаблонам об'єктів нарізки. Запускається цикл, який по черзі перевіряє наявність конфігурації марок поточного шаблону серед знайдених на фотографії різальної поверхні.

Сам файл шаблону розпізнавання представляє собою файл формату xml, з інформацією про координати реєстраційних марок відносно деякої точки відліку у міліметрах. Точкою відліку зазвичай приймається один з кутів об'єкта, що нарізується, та не має принципового значення, адже для розпізнавання важлива конфігурація – взаємне розташування реєстраційних марок. Вміст типового шаблону розпізнавання, що описує об'єкт промаркований сімома марками показано на рисунку 5.2.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <Job Name="OCC_agr-20171206-133238">
3    <Material Name="Faltkarton" Description="100g - 500g">
4      <MaterialSettings Thickness="0.5" Length="500" Width="700">
5        </MaterialSettings>
6      </Material>
7    <Geometry>
8      <Point X="470.000" Y="669.999"></Point>
9      <Point X="470.000" Y="349.999"></Point>
10     <Point X="170.000" Y="669.999"></Point>
11     <Point X="170.000" Y="29.999"></Point>
12     <Point X="30.000" Y="469.999"></Point>
13     <Point X="30.000" Y="239.999"></Point>
14     <Point X="470.000" Y="29.999"></Point>
15   </Geometry>
16 </Job>

```

Рисунок 5.2 – Вміст файлу з шаблоном об'єкта нарізки

Окрім інформації про реєстраційні марки, файл може містити теги про назву об'єкта, матеріал, з якого він зроблений, товщину та інші параметри, які характеризують об'єкт нарізки та визначають спосіб його нарізання. Утім нашу систему комп'ютерного зору цікавлять тільки назва об'єкта як ідентифікатор, координати марок та товщина матеріалу.

Процедура розпізнавання шаблону об'єкта нарізки серед марок виглядає наступним чином. На вхід приймається цільовий шаблон та набір піксельних координат знайдених на знімку марок. Також в якості аргументу приймається контейнер-словник вигляду «точка – набір ребер», який кожній знайдений на фотографії реєстраційній марці ставить у відповідність список ребер, що виходять з неї. Вміст цього словника формується одразу після завершення пошуку реєстраційних марок та необхідний для уникнення повторного перерахунку відстаней між марками при частій необхідності цю відстань отримати. Процедура, що відповідає за це, також паралельно формує ще один словник, який має вигляд «пара точок – відстань» і призначений для зберігання та швидкого доступу до відстані між будь-якими двома марками. Усі ці приготування є необхідними, оскільки зберігають дорогоцінний час при виконанні ітерацій процедури пошуку шаблону об'єкта серед марок, адже час виконання операції доступу до даних словника за ключем значно менший за час розрахунку відстані між точками на фото у міліметрах.

У першу чергу перевіряється, чи достатньо знайдено марок на фото. Якщо всього знайдених марок менше, ніж половина від кількості марок у шаблоні об'єкта, то процедура переривається, вважаючи, що об'єктів з таким шаблоном на знімку різальної поверхні немає. У такому разі процедура запускається з наступним шаблоном пошуку, якщо він є. Якщо ж марок замало для всіх переданих системі шаблонів, але описувані шаблонами об'єкти присутні на знімку різальної поверхні, то програма у будь-якому разі відмітить результати пошуку на копії знімка, виділивши знайдені марки і користувач зможе зробити висновки, створити ще один зразок марки для пошуку та запустити розпізнавання повторно.

Далі циклічно перебираються всі марки шаблону. Для кожної марки шаблону запускається ще один внутрішній цикл, який проходить по всім знайденим на фото маркам та перевіряє їх відповідність поточній марці шаблону за допомогою порівняння їх наборів ребер. Якщо марка знайдена на знімку має більше 50% ребер, що відповідають марці з шаблону, вона заноситься до словника пар відповідностей вигляду: «марка шаблону – марка на фотографії» та видаляється з набору знайдених марок. Перебір продовжується

далі, оскільки на різальній поверхні може бути одночасно кілька однакових об'єктів нарізки, тож одній марці з шаблону може відповідати декілька марок зі знімку.

Слід зазначити, що порівняння ребер відбувається з допуском у 3% від довжини ребра шаблонної марки, адже повної рівності довжин відповідних ребер бути не може через присутність допустимої похибки в системі позиціонування (приблизно  $\pm 1$  мм).

Після закінчення циклу пошуку марок шаблону серед знайдених на знімку методом порівняння наборів ребер, маємо заповнений словник їх відповідностей. Якщо кількість пар в словнику менша, ніж половина від кількості марок у цільовому шаблоні, процедура закінчується, оскільки це означає, що об'єктів з таким шаблоном на різальній поверхні немає. Якщо ж пар в словнику достатньо, то починається пошук вже окремих екземплярів заданого шаблону об'єкта серед відібраних пар марок.

Береться довільна (перша) *опорна пара* зі словника «марка шаблону – марка на фотографії» та видаляється з нього. Далі програма намагається знайти відносно неї всі інші марки, які разом сформують об'єкт нарізки. Одразу ж створюється новий програмний екземпляр об'єкта, який містить контейнер для його марок та поле ідентифікатора шаблону нарізки. До списку його марок заноситься позиція марки на фото з опорної пари. Також йому присвоюється ідентифікатор поточного шаблону. Після цього починається цикл перебору ребер шаблонної марки з опорної пари, всередині якого знаходиться вкладений цикл, який проходить по всім парам словника. У внутрішньому циклі для шаблонної марки поточної пари розраховується відстань від неї до шаблонної марки опорної пари та порівнюється з поточним ребром зовнішнього циклу. Якщо вони рівні, то розраховується відстань від марки на фото з поточної пари до марки на фото з опорної пари. Якщо ця відстань приблизно (з урахуванням 3% допуску) рівна значенню поточного ребра, а шаблонна марка – та сама, до якої направлене ребро першої марки, то поточна пара є ще однією маркою об'єкта нарізки і значення марки на фото додається до списку марок екземпляра цього об'єкта.

По завершенню циклів реконструкції структури об'єкта нарізки, залишається лише переконатись, що він точно відповідає шаблону об'єкта нарізки, адже до цього

часу перевірялись лише ребра, без урахування просторового розміщення точок. Звичайно, для об'єктів з великою кількістю марок та їх складною конфігурацією повна відповідність ребер до шаблону може слугувати вичерпним доказом того, що об'єкт розпізнано, але слід пам'ятати про те, що до списку потрапили також фальшиві марки, які утворюють потрібні ребра, але просторово розміщені не там, де це потрібно.

Саме тут на допомогу приходить алгоритм узгодженого дрейфу точок, який прийнявши на вхід список відмасштабованих координат марок шаблону та список координат марок на фото, поверне ще один список координат, порівнявши їх з тими, що відповідають фотографії, можна буде зробити висновок про те, чи був об'єкт розпізнаний, чи ні. Простими словами, алгоритм просто намагається «покласти» шаблонні точки на поверхню, де розташовані марки з фотографії так, щоб марки накладались максимально тісно. Повертаються саме оптимальні позиції накладених марок. Саме тому необхідно проводити масштабування координат шаблону так, щоб відстані між марками шаблону стали такими ж, як і відстані на фото.

Якщо об'єкт розпізнано успішно, то його програмний екземпляр додається у контейнер для подальшого повернення процедурою, а всі його марки видаляються зі словника пар «марка шаблону – марка на фотографії», оскільки не повинні брати участь у подальшому розпізнаванні. У випадку, коли алгоритм узгодженого дрейфу точок не підтвердив ідентифікацію об'єкта, словник залишається незмінним за винятком опорної точки об'єкта, яка була видалена на самому початку. Якщо в словнику після цього залишається замало пар (або на залишається зовсім), то цикл закінчується і процедура повертає колекцію ідентифікованих об'єктів. Інакше, обирається нова опорна пара та прохід циклу ідентифікації починається з початку.

Розпізнані об'єкти, у свою чергу, додаються до контейнера, що містить усі ідентифіковані об'єкти різних шаблонів, адже на різальній поверхні одночасно можуть бути розміщені представники різних шаблонів нарізки. Повна реалізація процедури ідентифікації об'єктів та допоміжного функціоналу наведено у додатку А.

Після завершення процедури розпізнавання, програма повертає колекцію, де зібрані всі розпізнані об'єкти – нарізки з їх унікальними ідентифікаторами та координатами марок в міліметрах. Форма даних, що повертає СКЗ буде залежати від того, у якому вигляді їх приймає на вхід комп'ютер різального станка (файл, пакети даних тощо).

Отримані дані несуть вичерпну інформацію про тип та положення об'єктів нарізки на робочій поверхні плоттера, що дозволять йому задати правильну траєкторію руху різального механізму уздовж контурів розпізнаних об'єктів.

### 5.3 Дослідження варіантів використання методу узгодженого дрейфу точок з різними налаштуваннями

Оскільки алгоритм узгодженого дрейфу точок має реалізації як для гнучкого (non-rigid), так і для жорсткого (rigid) співставлення наборів точок, необхідно обрати оптимальний для нашого випадку варіант.

В умовах різних координатних систем (на фото вісь ординат спрямована вниз, а точки шаблонів подано у традиційній декартовій системі координат з ординатою вгору) та одиниць вимірювання (пікселі та міліметри), на перший погляд, очевидним виглядає вибір гнучкої моделі узгодження множин точок.

Проте жорстке узгодження також має свої переваги, які не слід недооцінювати. До того ж, маючи отримане після калібрування співвідношення пікселів на фото до реальної відстані на робочій поверхні, можна легко змінити масштаб та перевести до нової системи координат точки шаблону.

Для вирішення питання найбільш доцільного методу узгодження, необхідно провести серію дослідів, змінюючи параметри алгоритму та якість вхідних даних. Серед параметрів CPD, доступних для зміни та вагомих для результатів дослідів є наступні:

- Максимальна кількість ітерацій  $N_{max}$ ;
- Ваговий коефіцієнт сторонніх точок  $w$ ;
- Коефіцієнт  $\sigma$ , що слугує для задання границь компонентів Гауссових моделей змішування;

- Допуск  $t$ , який разом з максимальною кількістю ітерацій визначає умови припинення роботи алгоритму.

Доцільною є лише зміна коефіцієнтів  $w$  та  $\sigma$ , у той час як умови припинення алгоритму змінювати не варто, оскільки це може призвести до його сповільнення або навпаки – скорочення роботи з втратами у якості. Кількість ітерацій за замовчуванням встановлена 150, а  $t = 10^{-6}$  і показує що в ході ітерації при величині зміни значень внутрішніх параметрів алгоритму, меншій ніж  $t$ , алгоритм досяг найкращого з можливих результатів.

На вхід подаватимемо два знімки промаркованих об'єктів нарізки з різною кількістю та конфігурацією РМ. Перший дослідний зразок об'єкта розпізнавання з сімома марками подано на рисунку 5.3.



Рисунок 5.3 – Простий об'єкт розпізнавання із сімома реєстраційними марками

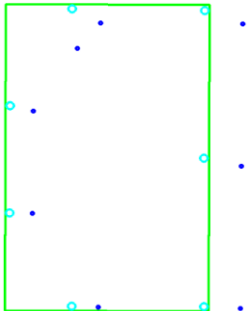
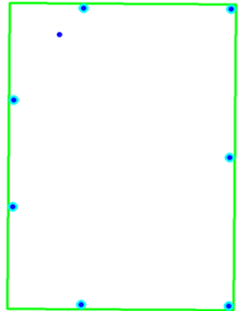
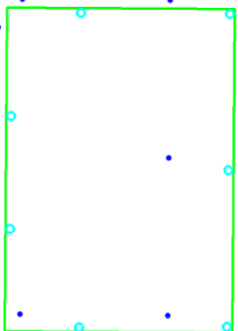
На виході розпізнавання будемо вимірювати наступні параметри:

1. Правильність узгодження – тобто, чи сумістились два набори точок у результаті роботи алгоритму. Якщо алгоритм дав неправильний результат, то всі інші показники значення не мають.
2. Точність узгодження – для цього будемо підраховувати відстані між відповідними точками з наборів, та порівнювати їх суми  $\Sigma$ .
3. Швидкість – кількість ітерацій  $N$  та час роботи алгоритму  $T$  у мікросекундах.

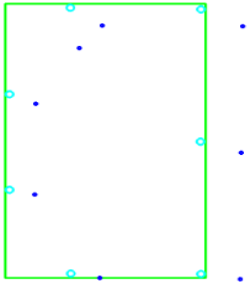
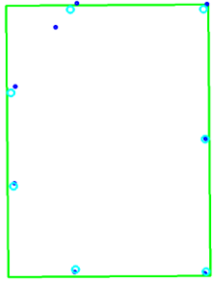
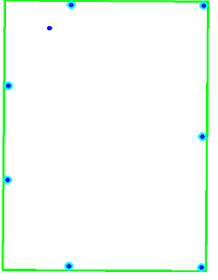


Спочатку підберемо найкращі коефіцієнти для максимально швидкого розпізнавання об'єкта гнучким узгодженням. Результати тестів відповідно до різних налаштувань алгоритму, разом зі схематичним зображенням результатів розпізнавання, наведено в таблиці 5.1.

Таблиця 5.1 – результати тестів алгоритму гнучкого узгодження CPD на об'єкті з 7 реєстраційними марками.

№ досліджу	$w$	$\sigma$	Результат	$\Sigma, px$	$N$	$T, мкс$	Положення точок
1	0.1	0	-	1088	16	9614	
2	0.2	0	+	0	26	15192	
3	1	0	-	2027	1	1016	

Продовження таблиці 5.1

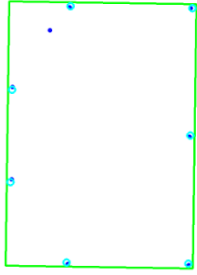
4	0.05	0	-	1252	17	10123	
5	0.1	1	+	171	10	6569	
6	0.1	2	+	0	6	4225	
7	0.1	3	+	0	6	4127	Як в досліді 6
8	0.1	4	+	0	6	4192	Як в досліді 6
9	0.2	2	+	0	6	4393	Як в досліді 6
10	0.05	2	+	0	6	4127	Як в досліді 6

На ілюстраціях темним кольором показано фактичне розташування марок, а світлим – позиції, що видав на виході алгоритм CPD, а також обмежувальний прямокутник, що означає межі об'єкта. Сірим кольором виділено найкращий результат за кількістю ітерацій та часом роботи. Як видно з результатів експериментів, параметр  $\sigma$  відіграє ключову роль у регулюванні точності та швидкості роботи алгоритму CPD.

Не дивлячись на те, що точки шаблону були відмасштабовані лише приблизно, щоб відстані між ними були максимально подібні до відстаней між точками на знімку, у результаті маємо бездоганне накладання у найкращих випадках. Це пояснюється особливостями гнучкого узгодження, яке передбачає нелінійні перетворення для досягнення найкращої відповідності двох наборів точок. Також бачимо, що алгоритм CPD ігнорує зайві точки, яких всього було знайдено 4 (на малюнках видна лише одна), що є дуже корисним у разі подальшого помилкового потрапляння до вибірки точок, що не є реєстраційними марками або належать іншим об'єктам.

Тепер подивимось на результати алгоритму в режимі жорсткого узгодження. Результати експериментів наведено у таблиці 5.2.

Таблиця 5.2 – результати тестів алгоритму жорсткого узгодження CPD на об'єкті з 7 реєстраційними марками.

№ дослідю	$w$	$\sigma$	Результат	$\Sigma$ , рх	$N$	$T$ , мкс	Положення точок
1	0.1	3	+	78	4	2202	
2	0.1	2	+	78	4	2090	Як в досліді 1
3	0.1	1	+	78	4	2281	Як в досліді 1
4	0.2	2	+	78	4	3173	Як в досліді 1
5	0.05	2	+	78	4	2117	Як в досліді 1

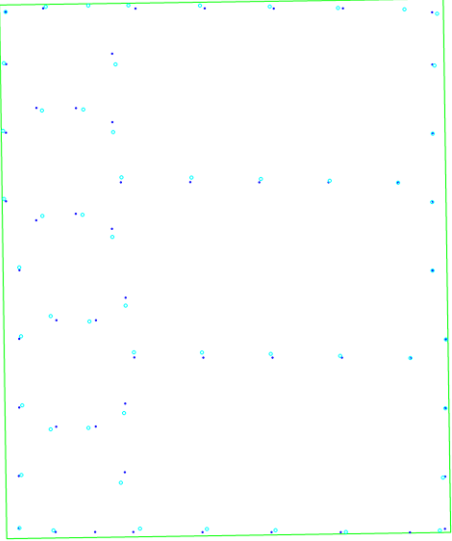
Жорстке узгодження показало двократний приріст швидкості у порівнянні з гнучким, але трохи поступається в точності, оскільки масштабування точок шаблону не може ідеально перевести відстані між ними з *мм* у *px*. Тепер перевіримо роботу обох варіантів CPD на більш складному об'єкті розпізнавання з 54 марками. Його зображення подано на рисунку 5.4.



Рисунок 5.4 – Складний об'єкт розпізнавання з 54 марками

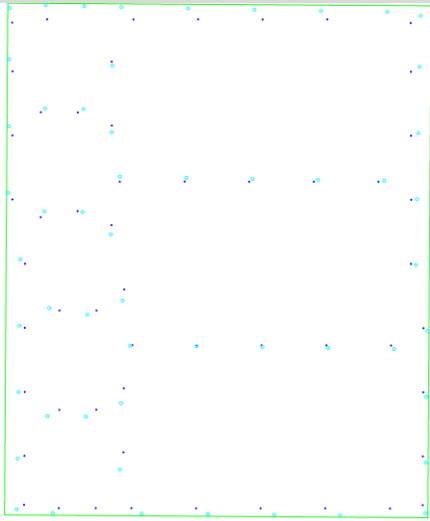
Результати дослідження застосування гнучкого узгодження точок для складного об'єкта розпізнавання наведено в таблиці 5.3.

Таблиця 5.3 – Результати застосування гнучкого узгодження до складного об'єкта

№ дослідю	$w$	$\sigma$	Результат	$\Sigma$ , рх	$N$	$T$ , мкс	Положення точок
1	0.1	3	+	1860	11	187433	
2	0.1	2	+	1860	10	172016	Як в досліді 1
3	0.1	1	+	1860	9	154824	Як в досліді 1
4	0.1	0.5	+	1860	13	222849	Як в досліді 1

Як бачимо, навіть гнучке перетворення не забезпечує бездоганного накладання марок, коли це стосується великого об'єкта з багатьма РМ. Результати, продемонстровані методом жорсткого узгодження, подані в таблиці 5.4.

Таблиця 5.4 – Результати застосування жорсткого узгодження до складного об'єкта

№ дослідю	$w$	$\sigma$	Результат	$\Sigma$ , px	$N$	$T$ , мкс	Положення точок
1	0.1	2	+	2036	5	38043	
2	0.1	1	+	2036	5	38621	Як в досліді 1
3	0.1	3	+	2036	6	44446	Як в досліді 1

Підсумовуючи проведені дослідження, можна впевнено сказати, що жорстке узгодження точок є більш ефективним, аніж гнучке, оскільки виконується за вдвічі-втричі менший час, виконуючи при цьому свою головну задачу в контексті розроблюваної системи комп'ютерного зору – підтвердження ідентифікації об'єктів.

Додатково слід зазначити, що для точних результатів розпізнавання повернутих довкола своєї осі об'єктів необхідно мати шаблони в мінімум чотирьох екземплярах – з кутами повороту кратними 90 градусам. Ця необхідність обумовлена нестабільною роботою алгоритму CPD при внесенні повороту в узгоджувані точки. Проблема стосується як жорсткого, так і гнучкого методів узгодження, але легко усувається при застосуванні повернутих шаблонів. Найбільшу стабільність CPD демонструє при повороті узгоджуваних точок у межах  $\pm 90^0$  відносно точок шаблону.

## 6 ДЕМОНСТРАЦІЙНА ПРОГРАМА

Для демонстрації можливостей розробленої системи комп'ютерного зору було розроблено програму з графічним інтерфейсом, яка виступає в ролі надбудови над ядром системи. Графічний інтерфейс програми складається з єдиного вікна, яке зображено на рисунку 7.1.

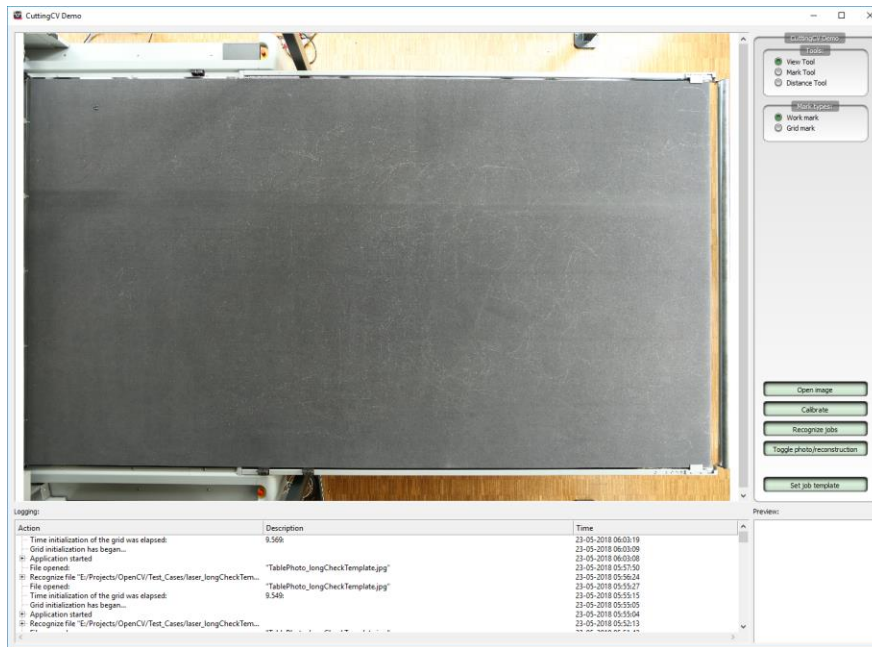


Рисунок 6.1 – Графічний інтерфейс демонстраційної програми

За основу для побудови зручного інтерфейсу користувача, було взято вигляд віджета з документації Qt [16], призначеного для демонстрації функціоналу Qt, що стосується афінних перетворень. Вікно демонстраційної програми складається з наступних елементів:

1. Зона перегляду поточного знімку, з яким працює система комп'ютерного зору, – займає 70% всього вікна.
2. Панель керування.
3. Звітна панель для відображення виконаних дій у формі деревоподібної структури.
4. Невеличке віконце для перегляду знайдених на головному зображенні за допомогою спеціального інструмента реєстраційних марок.

У зоні перегляду відображається фотознімок, з яким у поточний момент працює система комп'ютерного зору. Ця зона є інтерактивною та сприймає команди миші, дозволяючи масштабувати зображення та змішувати відображуваний фрагмент за допомогою полос прокрутки.

Окрім відображення поточного знімку, на зоні перегляду також з'являються текстові повідомлення про поточний статус виконуваного завдання, наприклад, кількість знайдених марок, етап калібрування або розпізнавання об'єктів. У якості індикатора статусу очікування під час виконання будь-якої операції на зоні перегляду відображається спеціальна анімація, що віддалено нагадує шестерню, що крутиться.

Справа від зони перегляду розташована стек-панель керування з перемикачами режимів взаємодії з зображенням та кнопками для подачі команд системі (рисунок 6.2). Усього режимів роботи з зображенням – три: перегляд (View Tool), пошук шаблонів марок (Mark Tool) та пошук відстані (Distance Tool).

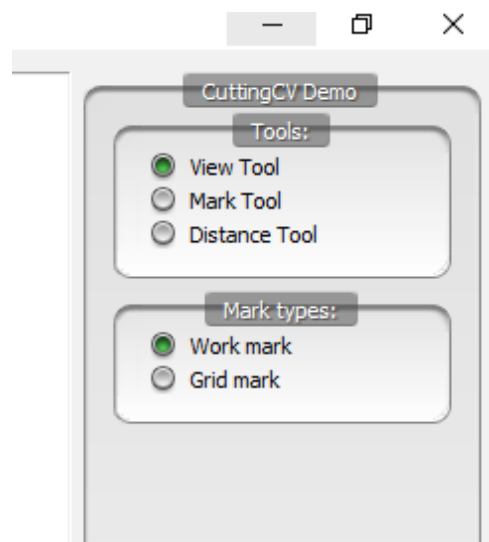


Рисунок 6.2 – Область перемикачів режимів роботи з екраном

У режимі перегляду зона перегляду буде реагувати лише на прокрутку колеса миші, масштабуючи відображуване на ньому зображення. У режимі пошуку марок, кліком лівої кнопки миші на зображенні, запускається процес пошуку марки в районі точки, на яку прийшовся клік. Якщо марка була знайдена, створюється її зразок та додається у



базу зразків марок системи, а також сама розпізнана марка відображається у віконці перегляду знизу справа. Віконце перегляду не інтерактивне і слугує лише для відображення знайдених інструментом пошуку марок.

Нижче перемикачів режимів знаходяться перемикачі типу створюваних зразків марок. Усього існує 2 типи зразків марок – марки калібрувальної сітки (grid mark) та робочі марки (work mark). До перших відносяться зразки, які використовуються тільки при пошуку марок калібрувальних сіток і ніколи не використовуються для розпізнавання марок об'єктів нарізки. І навпаки – робочі марки використовуються для розпізнавання будь-яких об'єктів окрім калібрувальних патернів. Такий розподіл підвищує ефективність використання шаблонів, оскільки застосування всіх наявних в системі шаблонів для різних цілей займало б забагато часу при пошуку марок. До того ж, калібрувальні марки зазвичай мають більший радіус, ніж робочі.

Третій режим роботи із зображенням призначений для цілей тестування системи позиціонування і дає можливість визначати реальні відстані між точками на зображенні. Після переходу в цей режим достатньо клікнути по черзі у двох точках знімку, після чого на звітній панелі під зображенням висвітиться відстань між зображеними точками у міліметрах. Цей режим працює тільки при виконаному попередньо калібруванні системи комп'ютерного зору.

Перемикачі режимів знаходяться у верхній частині панелі інструментів, а в нижній її частині розташовані кнопки основних команд для системи комп'ютерного зору. Кнопкова частина панелі керування містить всього 5 кнопок і зображена на рисунку 6.3.

Перша кнопка «Open image» викликає вікно стандартного провідника файлової системи для пошуку в ній зображення для відкриття. Обране зображення завантажується в систему комп'ютерного зору та відображається в зоні перегляду. При відкритті зображення не піддається ніякій обробці і відображається таким, яким є. Система працює з копією і всі зміни, що стосуватимуться зображення, не відобразяться на файлі, що був завантажений.

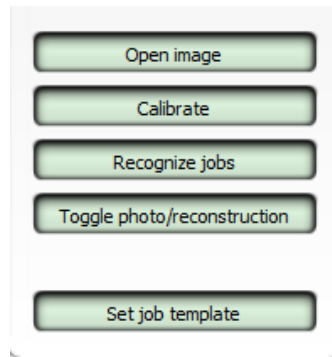


Рисунок 6.3 – Кнопки керування системою комп’ютерного зору

Друга кнопка «Calibrate» призначена для отримання системою калібрувальних даних. Після її натискання, як і для кнопки Open image, відкривається провідник, де користувач має вибрати калібрувальний знімок. Після вибору знімка починається власне процес калібрування. Після отримання калібрувальної інформації система зберігає в спеціальну папку копію калібрувального знімка та файл з координатами знайдених на ньому вузлів калібрувального патерну. При наступному запуску СКЗ ці дані зчитуються, та на їх основі відновляються калібрувальні матриці та система позиціонування.

Одним з недоліків розробленого інтерфейсу є необхідність вручну заносити в файл ініціалізації системи дані, які стосуються калібрування – розміри (кількість стовпців і рядків) та крок сітки, але це зберігає час на виконання калібрування, адже відпадає необхідність у додатковому аналізі зображення для з’ясування розмірів калібрувального патерну. Також, для узгодження системи позиціонування в папку, де зберігаються калібрувальні дані, слід помістити файл з координатами вузлів калібрувальної сітки відносно точки відліку різця. Якщо цього не зробити, то система буде рахувати від нижнього лівого краю калібрувальної сітки.

Після кнопки калібрування йде кнопка команди розпізнавання «Recognize jobs», яка починає процес розпізнавання об’єктів нарізки на знімку. Необхідною умовою розпізнавання є наявність у системі комп’ютерного зору калібрувальних даних, зразків РМ для пошуку на зображенні, а також мінімум одного шаблону об’єкта для розпізнавання. Після завершення процедури розпізнавання на екран виводиться оброблений знімок, на

якому виділені зеленою рамкою розпізнані об'єкти та обведені синіми колами знайдені марки. Також система виводить на звітну панель результати процедури розпізнавання у формі списку знайдених об'єктів, у кожного з яких є список координат його марок на знімку, їх координат відносно точки відліку системи та координат сформованих алгоритмом CPD. Також для кожного об'єкта зазначається шаблон, якому він відповідає. Відображення типового результату розпізнавання на звітній панелі зображено на рисунку 6.4.

Action	Description	Time
Recognize file "E:/Projects/OpenCV/Test_Cases/45DegProblem/RMR..."		22-05-2018 04:10:06
Found page 0:	coincided with .zcc:	
CPD reconstructed points (mm)		
Points relating origin (mm)		
Points on photo (pixels)		
Point 0:	[0, 0]	
Point 1:	[2821, 3496]	
Point 2:	[1803, 3476]	
Point 3:	[2875, 4531]	
Point 4:	[1912, 4040]	

Рисунок 6.4 – Вивід на звітну панель інформації про розпізнаний об'єкт

Також після розпізнавання за допомогою наступної кнопки «Toggle photo/reconstruction» можна подивитись на реконструйоване положення марок відносно точки відліку системи, а також результат роботи алгоритму CPD. Ця кнопка працює як перемикач між робочим зображенням та зображенням з реконструйованими положеннями марок. Ця кнопка, як і реконструйоване зображення, слугує лише в демонстраційних та тестових цілях, і ніяк не пов'язана з функціоналом СКЗ. Суть реконструйованого зображення у тому, що воно побудовано у масштабі 1px : 1мм, та не містить нічого зайвого окрім відмічених положень марок та точки відліку.

Остання кнопка називається «Set job template» і служить для завантаження файлів шаблонів об'єктів нарізки в СКЗ. Так само як і для зображень, завантаження відбувається через провідник файлової системи.

Не дивлячись на свою простоту, такий інтерфейс повністю охоплює функціонал розробленої СКЗ та підходить як для демонстраційних, так і для тестових цілей. Принцип його побудови залишає достатньо місця для нових елементів управління у разі появи нового функціоналу.

## 7 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

### 7.1 Опис ідеї проекту

Ідея стартап-проекту полягає у виході на ринок цифрової нарізки з революційною технологією. Ціллю проекту є заключення партнерського договору з однією або кількома провідними виробниками плоттерів для отримання фінансування подальших розробок та впровадження розробленої системи комп'ютерного зору в якості інтегрованого модуля окремої лінійки плоттерів для серійної нарізки.

Таблиця 7.1 – Опис ідеї стартап-проекту

<i>Зміст ідеї</i>	<i>Напрямки застосування</i>	<i>Вигоди для користувача</i>
	1. Промислова нарізка	Багатократний приріст швидкості, автоматизація процесу розпізнавання для кожної окремої ітерації нарізки, збільшення об'єму нарізаної продукції у результаті, зменшення зносу деталей ріжучого механізму плоттера, простота у використанні та налаштуванні
	2. Інженерна нарізка	Збереження високої точності ( $\pm 1$ мм) при економії часу завдяки потужним алгоритмам 2-D та 3-D реконструкції. Одноразове калібрування системи за умови незмінного положення камери

Визначимо сильні та слабкі сторони проекту та проведемо порівняльний аналіз з провідними фірмами-виробниками обладнання для цифрової нарізки.

Таблиця 7.2 – Сильні, слабкі та нейтральні сторони проекту

№ п/п	Техніко- економі- чні хара- ктерис- тики ідеї	(Потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна)	S (сильна сторона)
		Мій проект	Zund	Esko			
1.	Техноло- гічна со- бівар- тість	Собівар- тість плоттера +2000\$ собівар- тості СКЗ	Собівар- тість плоттера	Собівар- тість плот- тера	W За рахунок збіль- шення со- бівартості		
2.	Камери- детек- тори	2	1	1		N	
3.	Необхід- ність по- перед- нього ка- лібру- вання пе- ред вико- ристан- ням	Так	Ні	Ні	W Необхід- ність опе- ратору ви- конувати додаткові дії		

## Продовження таблиці 7.2

4.	Швид- кість роз- пізна- вання	Прямо пропор- ційна кі- лькості об'єктів розпізна- вання з <b>НИЗЬКИМ</b> значен- ням кое- фіцієнту пропор- ційності	Прямо пропор- ційна кі- лькості об'єктів розпізна- вання з <b>ВИСОКИМ</b> значен- ням кое- фіцієнту пропор- ційності	Прямо пропор- ційна кіль- кості об'єктів розпізна- вання з <b>ВИ-</b> <b>СОКИМ</b> зна- ченням ко- ефіцієнту пропорцій- ності			S  Значний виграш у швид- кості розпі- зна- вання
5.	Частота втру- чання людини в процес нарізки	Дуже рідко	Рідко	Рідко			S  Мінімі- зація людсь- кого втру- чання
6.	Рівень автома- тизації виробни- цтва	Високий	Середній	Середній			S  Вищий рівень автома- тизації

## 7.2 Технологічний аудит ідеї проекту

Для визначення здійсненності задуму проекту було проведено його ретельний аналіз. Результати подано у таблиці 7.3.

Таблиця 7.3 – Технологічна здійсненність ідеї проекту

<i>Ідея проекту</i>	<i>Технології її реалізації</i>	<i>Наявність технологій</i>	<i>Доступність технологій</i>
Програма що розпізнає та визначає положення об'єктів на різальній поверхні за допомогою фотографії	Технологія 1. Повністю самостійна розробка всіх компонентів програми без використання сторонніх бібліотек на мові C++.	Потребує розробки	Немає у відкритому доступі
	Технологія 2. Використання функціоналу відкритих бібліотек (OpenCV, CVV) комп'ютерного зору для стандартних методів обробки зображення та розпізнавання об'єктів на ньому	Наявні	У відкритому доступі з відкритим кодом
	Технологія 3. Покупка платних потужних бібліотек комп'ютерного зору для досягнення максимально високої ефективності з меншими витратами часу.	Наявні	Платні, з закритим кодом

*Обрана технологія реалізації ідеї проекту:* Технологія 2, оскільки вона не потребує початкових капіталовкладень та передбачає використання деяких готових рішень.



### 7.3 Аналіз ринкових можливостей запуску стартап-проекту

Проаналізуємо поточний стан ринку цифрової нарізки на предмет наявності попиту, обсягів та динаміки виробництва. Дані аналізу наведені в таблиці 7.4.

Таблиця 7.4 – Попередня характеристика потенційного ринку стартап-проекту

<i>№ n/n</i>	<i>Показники стану ринку (найменування)</i>	<i>Характеристика</i>
1.	Кількість головних гравців, од	3–4
2.	Загальний обсяг продаж, грн/ум.од	60-70 мільйонів євро
3.	Динаміка ринку (якісна оцінка)	Зростає
4.	Наявність обмежень для входу (вказати характер обмежень)	Обмеження на сфери застосування нової технології – підходить розпізнавання тільки промаркованих об'єктів нарізки
5.	Специфічні вимоги до стандартизації та сертифікації	відсутні
6.	Середня норма рентабельності в галузі (або по ринку), %	70%

Попередня оцінка показала, що ринок є досить привабливим для входження, хоча існують і деякі обмеження на нішу, яку розроблюваний продукт здатен зайняти у ньому. Розглянемо потенційні групи клієнтів, їх характеристики та вимоги до товару.

Таблиця 7.5 – Характеристика потенційних клієнтів стартап-проекту

<i>№ n/n</i>	<i>Потреба, що фор- мує ринок</i>	<i>Цільова аудито- рія (цільові сегме- нти ринку)</i>	<i>Відмінності у поведінці різних потенційних ці- льових груп кліє- нтів</i>	<i>Вимоги споживачів до то- вару</i>
1.	Нарізка різнома- нітних матеріа- лів	Текстильна та взуттєва промис- ловість, полігра- фія, дизайнерські студії, та будь-які інші установи, які мають потребу у промисловій та високоточній на- різці матеріалів	Форма і об'єми виробництва, ці- льовий матеріал, необхідність і можливість пе- реходу до циф- рової нарізки	- до продукції: безвідмовна робота, висока якість та шви- дкість нарізки, простота в експлуатації, максимальний рівень автоматизації при без- перервному виробництві  - до компанії-постачальника: гарантія якості, допомога у налаштуванні техніки відпо- відно до цілей використання та її гарантована технічна підтримка

Тепер на основі отриманих відомостей про потенційні групи клієнтів можна оцінити фактори, що сприяють – фактори можливостей ринку, та фактори, що заважають впровадженню проекту.

Таблиця 7.6 – Фактори загроз

<i>№ n/n</i>	<i>Фактор</i>	<i>Зміст загрози</i>	<i>Можлива реакція компанія</i>
1.	Фактор фірм-виробників плоттерів	Небажання фірм-виробників впроваджувати нову технологію	Пошук альтернативних клієнтів серед виробників цифрових різаків
2.	Фактор інертності клієнтів	Задоволеність поточним продуктом і небажання переходити до нового	Впровадження акцій обміну або модернізація попередніх версій устаткування за окрему доплату

Таблиця 7.7 – Фактори можливостей

<i>№ n/n</i>	<i>Фактор</i>	<i>Зміст можливості</i>	<i>Можлива реакція компанія</i>
1.	Фактор прогресу	Жодна фірма-виробник цифрових різаків не відмовиться вдосконалити свій продукт	Більш тісна співпраця з фірмами-виробниками задля подальшого вдосконалення програмного продукту
2.	Фактор користувачького досвіду	Зазвичай скарги і пропозиції користувачів стають імпульсом для подальшого розвитку продукції	Проведення широких досліджень користувачького досвіду та відгуків для обрання подальшого вектору розвитку програми

Після проведення аналізу конкурентного середовища ринку ЦНЗ отримали таблиці 7.8 та 7.9.

Таблиця 7.8 – Ступеневий аналіз конкуренції на ринку

<i>Особливості конкурентного середовища</i>	<i>У чому проявляється дана характеристика</i>	<i>Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможними)</i>
1. Вказати тип конкуренції - олігополія	Існує декілька компаній, що займаються цифровою нарізкою, з приблизно рівним обсягом продажів	Стимулює кожен компанію до пошуку революційних рішень
2. За рівнем конкурентної боротьби - міжнаціональний	Компанії-виробники локалізовані в різних країнах	Залучення національних ресурсів на підтримку та розвиток галузі в державі
3. За галузевою ознакою - внутрішньогалузева	Конкуренція зосереджена лише в галузі нарізки	Компанії цілковито зосереджуються на розвитку в рамках окремої галузі або намагаються отримати доступ до суміжних
4. Конкуренція за видами товарів: - товарно-видова	Випуск якомога більш багатоцільових різаків, що поєднують велику кількість параметрів налаштувань нарізки та здатність нарізати різноманітні матеріали	Постійне комбінування технологій у пошуках оптимального рішення

## Продовження таблиці 7.8

5. За характером конкурентних переваг - цінова і нецінова	Різні компанії роблять ставку на функціонал або на ціну продукції	Випуск широкого асортименту продукції з різним функціоналом і, відповідно, ціною
За інтенсивністю - марочна	Приблизно однакові ціни та функціонал продукції різних виробників	Пошук рішення, що призведе до отримання рішучої переваги над конкурентами

Таблиця 7.9 – Аналіз конкуренції в галузі за М. Портером

<i>Складові аналізу</i>	<i>Прямі конкуренти в галузі</i>	<i>Потенційні конкуренти</i>	<i>Постачальники</i>	<i>Клієнти</i>	<i>Товари-замінники</i>
	Zund, Esko, Dyss	Відсутність світового імені (реклама), відсутність інвестицій та матеріалу для досліджень	Постачальником в даному випадку буде партнерська фірма, що виготовляє різак	Наявність потреби в новій технології, форма застосування продукту	Можливість перехоплення ідеї вдосконалення різаків та більш швидке її впровадження

## Продовження таблиці 7.9

Висновки:	Середній рівень конкуренції	<ul style="list-style-type: none"> <li>- Можливість виходу на ринок через співпрацю з однією з існуючих компаній</li> <li>- Конкурентами стануть автоматично всі інші компанії, що відмовляються від співпраці</li> </ul>	У деякій мірі фірма-партнер диктує умови роботи, завдяки постачанню робочого матеріалу та вимогам	Клієнти диктують в основному цілі застосування цифрових різаків та цільовий матеріал нарізки	Цільова технологія може стати вирішальною в питанні домінування на ринку цифрових різаків, тож важливо впровадити її першими
-----------	-----------------------------	---	---	--	--

Існуюча конкурентна ситуація на ринку цифрової нарізки тільки сприятиме скорішому впровадженню нової технології, яку представляє розроблена система комп'ютерного зору. Жодна фірма-виробник плоттерів не відмовиться від технології, яка потенційно визначає майбутнє індустрії ЦНМ. Визначені фактори конкурентоспроможності стартап-проекту подано в таблиці 7.10.

Таблиця 7.10 – Обґрунтування факторів конкурентоспроможності

№ n/n	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Інноваційна ідея	Застосування нових революційних методів розпізнавання є значною перевагою на ринку

## Продовження таблиці 7.10

2.	Можливість розширення партнерської бази програми	У кінці кінців усі виробники та споживачі цифрових різаків звернуться до нової технології, отже, усі вони є потенційними партнерами
3.	Можливість переобладнання вже існуючих типів цифрових різаків	Конструктивна простота технології не вимагає запуску принципово нової лінійки продукції
4.	Додаткові витрати на модернізацію та фотообладнання	Для забезпечення високої якості розпізнавання конструкція має бути максимально стабільною та використовувати потужний фотоапарат з мінімальним спотворення зображення

Таблиця 7.11 – Порівняльний аналіз сильних та слабких сторін «Системи комп'ютерного зору для цифрової нарізки»

№ n/n	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з «Системою комп'ютерного зору для цифрової нарізки»						
			-3	-2	-1	0	+1	+2	+3
1.	Інноваційна політика	15		Dyss			Esco	Zund	
2.	Можливість розширення клієнтської бази програми	12				Esco Dyss	Zund		
3.	Можливість переобладнання вже існуючих типів цифрових різаків	14					Esco	Zund Dyss	

## Продовження таблиці 7.11

4.	Додаткові витрати на модернізацію та фото-обладнання	15					Zund Esco Dyss		
----	--	----	--	--	--	--	----------------------	--	--

Тепер проведемо SWOT-аналіз проекту та сформуємо матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities).

Таблиця 7.12 – SWOT – аналіз стартап-проекту

<p><b>Сильні сторони:</b></p> <p>Революційний метод розпізнавання, конструктивна простота, приріст в швидкості</p>	<p><b>Слабкі сторони:</b></p> <p>Збільшення ціни продукту, необхідність калібрування, надвисока чутливість до зміни положення камери</p>
<p><b>Можливості:</b></p> <p>Розширення клієнтської бази, виведення партнерської компанії на лідируючі позиції на ринку, співпраця з іншими компаніями</p>	<p><b>Загрози:</b></p> <p>Перехоплення ідеї конкурентами, більшість клієнтів може задовольняти статус кво та відсутність необхідності в перевагах, що надає нова технологія</p>

Маючи дані SWOT-аналізу, перейдемо до розробки альтернатив ринкової поведінки для виведення стартап-проекту на ринок та орієнтовний оптимальний час його ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.



Таблиця 7.13 – Альтернативи ринкового впровадження стартап-проекту

<i>№ п/п</i>	<i>Альтернатива (орієнтовний комплекс заходів) ринкової поведінки</i>	<i>Імовірність отримання ресурсів</i>	<i>Строки реалізації</i>
1.	Заклучення партнерського договору з фірмою-виробником плоттерів	100%	Від 6 до 12 місяців.
2.	Непряме дослідження існуючих прототипів різаків безпосередньо у кінцевих споживачів	80%	У 3-4 рази більші, ніж при співпраці з фірмою виробником
3.	Самостійна закупівля станків для тестування та розробки продукту	50% (залежить від наявності спонсорів)	Ті ж самі

Очевидно єдиним прийнятним варіантом виходу на ринок є заключення партнерського договору з існуючими фірмами виробниками. При виборі самостійної закупівлі обладнання передбачаються занадто великі витрати, які можливо доведеться покривати за рахунок кредитів. І, що більш небезпечно, так само як і при співпраці зі сторонніми власниками обладнання, втрачається стратегічний момент виходу на ринок при новизні технології.

#### 7.4 Розроблення ринкової стратегії проекту

Визначимо стратегію охоплення ринку. Опис цільових груп потенційних споживачів наведено в таблиці 7.14. Опис обраних варіантів базових стратегій розвитку та конкурентної поведінки подано у таблицях 7.15 та 7.16 відповідно.

Таблиця 7.14 – Вибір цільових груп потенційних споживачів

<i>№ n/n</i>	<i>Опис профілю цільової групи потенційних клі- єнтів</i>	<i>Готовність споживачів сприйняти продукт</i>	<i>Орієнтовний попит в межах цільової групи (сегменту)</i>	<i>Інтенсив- ність кон- курентії в сегменті</i>	<i>Простота входу у сегмент</i>
1.	Текстильна про- мисловість	Висока	Високий	Середня	Проблем не пе- редбачається
2.	Шкіряна проми- словість	Висока	Високий	Середня	Проблем не пе- редбачається
3.	Поліграфія	Висока	Високий	Висока	Проблем не пе- редбачається
4.	Дизайнерські студії	Низька	Низький	Низька	Не зацікавлені в перевагах но- вого продукту
<i>Які цільові групи обрано: текстильна та шкіряна промисловості, поліграфія</i>					

Таблиця 7.15 – Визначення базової стратегії розвитку

<i>№ n/n</i>	<i>Обрана аль- тернатива розвитку</i>	<i>Стратегія охоп- лення ринку</i>	<i>Ключові конкурентоспромо- жні позиції відповідно до об- раної альтернативи</i>	<i>Базова стра- тегія розвитку</i>
1.	Відсутність нової лінійки товару	Модернізація го- тової продукції	Відсутність додаткових ви- трат на виробництво нової лі- нійки продукції, мінімальний ризик, зондування ринку	Стратегія ліде- рства по витра- тах

## Продовження таблиці 7.15

2.	Запуск нової лінійки товару	Заохочення клієнтів сильними сторонами продукту	Сильні сторони продукту гарантують окупність стратегії	Стратегія диференціації
----	-----------------------------	---	--	-------------------------

Перейдемо до визначення базової стратегії конкурентної поведінки, яка визначить курс розвитку проекту на найближчі роки.

Таблиця 7.16 – Визначення базової стратегії конкурентної поведінки

<i>№ п/п</i>	<i>Чи є проект «першопроходьцем» на ринку?</i>	<i>Чи буде компанія шукати нових споживачів або забирати існуючих у конкурентів</i>	<i>Чи буде компанія копіювати основні характеристики товару конкурента і які?</i>	<i>Стратегія конкурентної поведінки</i>
1.	Так	Так	Ні	Стратегія лідера

Залишилося на основі вимог споживачі з обраних сегментів та в залежності від обраних базових стратегій розвитку та конкурентної поведінки, сформуванати стратегію позиціонування, яка визначить ринкову позицію продукту, за якою його будуть ідентифікувати споживачі.

Таблиця 7.17 – Визначення стратегії позиціонування

<i>Вимоги до товару цільової аудиторії</i>	<i>Базова стратегія розвитку</i>	<i>Ключові конкурентоспроможні позиції власного стартап-проекту</i>	<i>Вибір асоціацій, які мають сформувати комплексну позицію власного проекту</i>
Надійність, точність, довгострокова гарантія, швидкість	Стратегія диференціації	Новизна, необхідність, простота	Спеціалізація на широкому спектрі потреб (широка клієнтська база), можливість модернізації існуючої продукції, тісна співпраця з споживачами для вдосконалення продукції

## 7.5 Розроблення маркетингової програми стартап-проекту

Таблиця 7.18 – Визначення ключових переваг концепції потенційного товару

<i>№ п/п</i>	<i>Потреба</i>	<i>Вигода, яку пропонує товар</i>	<i>Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)</i>
1.	Швидка нарізка в конвеєрному режимі	Мінімальний час ідентифікації об'єктів на нарізальній поверхні	Великий виграш в швидкості розпізнавання
2.	Мінімум людського втручання	Максимальна автоматизація процесу розпізнавання та нарізки	При можливості, додати повністю автоматичний режим за таймером (період зміни об'єктів на нарізальній поверхні) або за розпізнаванням зміни стану нарізальної поверхні

Таблиця 7.19 – Опис трьох рівнів моделі проекту

<i>Рівні товару</i>	<i>Сутність та складові</i>		
I. Товар за задумом	Товар задовольняє потребу споживача в швидкій та точній цифровій нарізці матеріалів як за конвеєрним типом, так і за одиничним		
	Властивості / характеристики	М/Нм	Вр, Тх, Тл, Е, Ор
	1. Програма розпізнавання на мові С++ 2. Параметри різаків згідно з специфікаціями партнерської фірми виробника		
	Якість: стандарти партнерської фірми-виробника		
	Пакування		
	Марка: назва фірми виробника + програма «Система комп'ютерного зору для цифрової нарізки»		
	До продажу: гарантія, доставка, монтаж, калібрування		
	Після продажу: технічне обслуговування, консультація, оновлення ПЗ		
За рахунок чого потенційний товар буде захищено від копіювання: закритий програмний код системи			

Таблиця 7.20 – Визначення меж встановлення ціни

<i>Рівень цін на товари-замінники</i>	<i>Рівень цін на товари-аналоги</i>	<i>Рівень доходів цільової групи споживачів</i>	<i>Верхня та нижня межі встановлення ціни на товар/послугу</i>
50-70 тис. \$	80-150 тис. \$	У середньому 0.5-1 млн. \$ у рік	10 тис. \$ + ціна ризака (80-150 тис. \$)

Згідно специфіки закупівельної поведінки цільових клієнтів, а також з урахуванням функцій збуту, які нам треба буде виконувати як постачальникам товару, сформуємо прийнятну систему збуту.

Таблиця 7.21 – Формування системи збуту

<i>Специфіка закупівельної поведінки цільових клієнтів</i>	<i>Функції збуту, які має виконувати постачальник товару</i>	<i>Глибина каналу збуту</i>	<i>Оптимальна система збуту</i>
Невпевнена купівельна поведінка	Транспортування, зберігання, встановлення контакту, «поділ»	2	Вертикальна

Провівши всебічний аналіз розробленої продукції, можна сформувати стратегію поведінки у полі маркетингових комунікацій. Згідно специфіки поведінки цільових клієнтів та каналів комунікації, якими вони користуються, сформуємо відповідні завдання та концепції рекламного звернення.

Таблиця 7.22 – Концепція маркетингових комунікацій

<i>Специфіка поведінки цільових клієнтів</i>	<i>Канали комунікацій, якими користуються цільові клієнти</i>	<i>Ключові позиції, обрані для позиціонування</i>	<i>Завдання рекламного повідомлення</i>	<i>Концепція рекламного звернення</i>
Високі вимоги до продукту, постійний зв'язок з технічною підтримкою, закупівля обмеженої кількості одиниць товару (1-2)	Формальні канали комунікацій	Швидка, проста і незмінно точна нарізка	Донести всі переваги нового методу розпізнавання об'єктів на різальній поверхні	Необхідно донести вигідність та надійність використання передових технологій у сфері цифрової нарізки

**Висновок:**

Ринковий аналіз сфери цифрової нарізки показав, що ринок у цій сфері поступово розширюється і не має одноосібного лідера. У такій ситуації нова революційна технологія, на звання якої претендує розроблена система комп'ютерного зору, може на довгий строк принести вирішальну перевагу тій фірмі, яка прийме цей стартап-проект як партнерську програму для найшвидшого виводу на ринок.

Вихід на ринок найдоцільніше буде оформити як випуск нової лінійки плоттерів з новими можливостями за рахунок впровадженої технології. Далі можна додати можливість переобладнання попередніх версій станків для використання нової системи комп'ютерного зору. Подальший розвиток проекту буде цілком залежати від обраного партнерською фірмою курсу.

## ВИСНОВКИ

У результаті всіх виконаних в контексті даної магістерської дисертації робіт та досліджень було розроблено сучасну, ефективну та надійну систему комп'ютерного зору для серійної цифрової нарізки матеріалів промаркованих реєстраційними марками матеріалів.

Після проведення декомпозиції головної задачі розроблюваної системи було виконано ґрунтовне дослідження існуючих рішень у предметній області підзадач. Існуючий досвід вирішення подібних або суміжних проблем дозволив розробити на його основі рішення для конкретної задачі. Так, для створення правильної системи позиціонування було вирішено проблему боротьби з оптичними спотвореннями фотосистем, яка одночасно вирішує питання переходу від координат на фотографії до координат відносно ріжучого механізму плоттера.

Два найкращих інструменти пошуку реєстраційних марок на зображенні було вирішено скомбінувати так, щоб один доповнював інший. У результаті була отримана гнучка система пошуку реєстраційних марок на зображенні за зразком, яка передбачає створення зразків марок за допомогою наявного в розробленій системі інструменту, що потребує мінімуму людського втручання.

Найбільшим успіхом дослідження є розробка потужного способу ідентифікації об'єктів за шаблоном, який частково базується на елементах теорії графів та підкріплюється застосуванням одного з найсучасніших алгоритмів узгодження двох множин точок, що носить назву «узгодженого дрейфу точок».

У сукупності всі ці результати дозволили створити таку систему, яка може з точністю до міліметра визначити положення об'єкта на різальній поверхні плоттера за лічені секунди зберігаючи як час, так і ресурс механізму станка для нарізки, а не пошуку положення об'єкта для нарізки.



## РЕКОМЕНДАЦІЇ

На даному етапі розроблена в ході наукового дослідження система комп'ютерного зору вже може претендувати на вихід на ринок цифрової нарізки матеріалів. Оцінка технології, виконана у розділі присвяченому стартап-проекту, показала його теоретично-високу привабливість для інвестицій. Першочерговою задачею є пошук партнерів серед розробників плоттерів, таких як Zund, Esko, Dyss та інші провідні гравці на ринку цифрової нарізки. Усі вони зацікавлені у покращенні якості власної продукції за рахунок інноваційних рішень, а тому навряд відмовляться від вигідного для них партнерства.

Усі подальші вдосконалення системи вважаю доцільним проводити лише після отримання ТЗ та специфікації для конкретного обладнання, у яке потрібно буде інтегрувати дану систему комп'ютерного зору. Відкритим залишається питання протоколу обміну між системою комп'ютерного зору та операційною системою плоттера, керування глобальною камерою через її програмний інтерфейс, а також кінцеву форму програмного додатку у вигляді самостійної програми, сервісу тощо.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Zünd Systemtechnik AG [Електронний ресурс] – Режим доступу до ресурсу: <https://www.zund.com/>.
2. ISO/IEC 14882:2017 [Електронний ресурс] // International Organization for Standardization. – 2017. – Режим доступу до ресурсу: <https://www.iso.org/standard/68564.html>.
3. Документація OpenCV [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.opencv.org/>.
4. CUDA [Електронний ресурс] // OpenCV library. – 2018. – Режим доступу до ресурсу: <https://opencv.org/platforms/cuda.html>.
5. Є. А. Іофіс. Фотокінотехніка/ І. Ю. Шебалін. — Москва: «Советская энциклопедия», 1981. — 447 с.
6. Волосов Д. С. Фотографічна оптика – Москва: «Искусство», 1978. – 543 с.
7. Сивухін Д. В. Загальний курс фізики. Оптика / Д. В. Сивухін. – Москва: "Наука", 1985. – 752 с.
8. Р. Шеліський, Комп'ютерний зір: Алгоритми та Застосунки – Нью-Йорк: Springer, 2011. – 957 с.
9. С. Маллік. Знаходження точок за допомогою OpenCV [Електронний ресурс] // Вивчення OpenCV – 2015. – Режим доступу до ресурсу: <https://www.learnopencv.com/blob-detection-using-opencv-python-c/>.
10. Співставлення зразка [Електронний ресурс] // Документація OpenCV 2.4.13.3. – Режим доступу до ресурсу: [http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template\\_matching/template\\_matching.html](http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html).
11. Г. Р. Брадський, А. Келер. Вивчаючи OpenCV – Севастополь: O'Reilly Media, 2008. – 556 с.
12. Besl P. J. A Method for Registration of 3-D Shapes / P. J. Besl, N. D. McKay. // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1992. – С. 239 – 256.

13. New algorithms for 2D and 3D point matching: pose estimation and correspondence / [S. Gold, C. Lu, A. Rangarajan та ін.]. // Pattern Recognition. – 1998. – С. 1019–1031.
14. Tsin Y. A Correlation-Based Approach to Robust Point Set Registration / Y. Tsin, T. Kanade. // Computer Vision ECCV. – 2004. – С. 558–569.
15. Myronenko A. Point Set Registration: Coherent Point Drift / A. Myronenko, X. Song. // IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. – 2010. – С. 2262–2275.
16. Affine Transformations [Електронний ресурс] // Qt Documentation – Режим доступу до ресурсу: <http://doc.qt.io/qt-5/qtwidgets-painting-affine-example.html>.