

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”

М.Р. МЕЛЬНИК

ЛАБОРАТОРНИЙ ПРАКТИКУМ

з курсу “Розпізнавання образів та комп’ютерний зір”

Рекомендовано до друку Науково-методичною радою
Національного університету “Львівська політехніка”
як навчальний посібник для студентів
спеціальності F3 «Комп’ютерні науки»

Видавництво Національного університету “Львівська політехніка”

Львів – 2025

ББК
УДК 004.414.2;004.942

Рекомендовано до друку Науково-методичною радою
Національного університету «Львівська політехніка»
як навчальний посібник для студентів
базового напрямку F3 «Комп'ютерні науки»
(протокол № ____ від ____ _____ 2025р.)

Рецензенти Цмоць І.Г., д.т.н., проф. каф. АСУ, НУ «Львівська політехніка»
Русин Б.П. – д.т.н., проф., зав. відділу методів і систем обробки та ідентифікації
зображень, ФМІ НАН України

ЗМІСТ

1. ТЕОРЕТИЧНА ЧАСТИНА.....	4
1.1 Методи розпізнавання образів.....	4
1.2 Лінійне стеження	12
1.3 Метод Яскравої точки	16
1.4 Виділення об'єкту на зображенні	19
1.5 Підготовка зображення до розпізнавання	21
1.6 Сегментація рухомих об'єктів у відео-поточі	24
2. ПРАКТИЧНА ЧАСТИНА	34
2.1 Алгоритм стабілізації образу.....	34
2.2 Алгоритм знаходження фону зображення	39
2.3 Трекінг (супроводження) об'єктів	44
2.4 Особливості реалізації в системі MatLab	48
3. ЛАБОРАТОРНА РОБОТА № 1	54
4. ЛАБОРАТОРНА РОБОТА № 2	54
5. ЛАБОРАТОРНА РОБОТА № 3	54
ЛІТЕРАТУРА.....	56

1. ТЕОРЕТИЧНА ЧАСТИНА

1.1 Методи розпізнавання образів

В залежності від постановки задачі розрізняють:

1. Роботи евристичного напрямку, для яких характерно знайти практично придатний алгоритм розпізнавання, базуючись на інтуїтивних міркуваннях.
2. Роботи математичного напрямку, в основу яких покладені деякі гіпотези про розв'язуючі функції.
3. Роботи лінгвістичного напрямку.

Коли розв'язок залежить від зображення, яке потрібно розпізнати, і є функцією від компоненти дискретного представлення або взагалі від значень ознак, тоді цю функцію називають розв'язуючою функцією.

Розв'язуюче правило — коли кожному зображенню ставлять у відповідність визначене найменування класу.

Щоб знайти розв'язок, що задовольняє всім потребам, потрібно зробити над компонентами зображення визначену послідовність обчислювальних і логічних операцій. Ця послідовність називається алгоритмом розпізнавання. Ідея, яка покладена в основу алгоритму розпізнавання, називається методом розпізнавання. Розв'язуюча функція задає розбиття множини V всіх значень сигналу W на підмножини. Ці підмножини називаються областями розв'язків. Для всіх значень сигналу, які належать одній області розв'язків, розв'язуюча функція має постійне значення. Інколи зручно задавати розв'язуючу функцію за допомогою так званих дискримінантних функцій, які визначені для кожної області розв'язку. В випадках, коли існує істинна класифікація і розв'язок, прийнятий за клас представленого зображення, не збігається з його істинним класом, говорять про помилку розпізнавання. Число помилок можна зменшити, якщо передбачити для розв'язуючого правила можливість видавати для сумнівних зображень невизначений розв'язок або так звану відмову від розпізнавання. Оскільки розв'язок при наявності випадкових спотворень є

випадковим, то говорять про імовірність помилки і імовірність відмови. Ці величини є найважливішими характеристиками системи розпізнавання або алгоритму розпізнавання.

Ймовірність помилки є основним критерієм, за яким оцінюється практична застосовність системи розпізнавання в задачах, де існує об'єктивна правильна класифікація. Тому проблема розпізнавання формулюється як знайдення оптимальних за цим критерієм розв'язуючих правил.

При розпізнаванні зображень, вільних від випадкових спотворень, але інваріантних до переносу, розв'язуюче правило може бути отримане за допомогою процедури центрування: деяка характерна точка зображення повинна бути шляхом переносу всього зображення поміщена в фіксовану точку поля зору. Після цього всі зображення, які відрізняються одне від одного тільки переносом, стають ідентичними і розв'язок може бути зроблений незалежно від початкового положення. Аналогічно може бути здійснена нормалізація зображення за їх розмірами. Але в умовах спотворень центрування і взагалі нормалізація є дуже складною.

1.1.1 Лінгвістичний (синтаксичний) метод

Якщо опис образів здійснюється за допомогою підобразів і їх співвідношень, то для конструювання системи розпізнавання використовують лінгвістичний або синтаксичний підхід з використанням принципу загальності властивостей. Основне припущення, яке робиться в цьому методі, ґрунтується на тому, що образи, які належать одному і тому ж класу, володіють рядом загальних властивостей або ознак, які відображають подібність таких образів. Ці загальні властивості можна частково ввести в пам'ять системи розпізнавання. Коли системі представити некласифікований образ, то вибирається набір ознак, що визначаються, причому останні інколи кодуються, а потім вони порівнюються з ознаками, закладеними в пам'яті системи розпізнавання. При використанні даного методу основна задача полягає у

виділенні загальних властивостей за набором образів, належність яких шуканому класу відома.

Вибір методу синтезу системи не розв'язує до кінця проблеми складання конкретної програми і проблеми реалізації. В більшості випадків є образи, що представляють кожний з розглядуваних класів. В таких випадках можна скористатися методами розпізнавання, які називаються "навчання з учителем". У схемі "навчання з учителем" система навчається розпізнавати образи за допомогою різного роду адаптивних схем. В деяких прикладних задачах для елементів, що належать до визначеного класу, навчаючі множини невідомі. У таких випадках, властиво, і можна звернутися до методів "розпізнавання з учителем". Розпізнавання по схемі навчання з учителем характеризується тим, що відома правильна класифікація кожного навчаючого образу. У випадку навчання без учителя потрібно конкретно вивчити класи образів, які є в даній інформації.

Типовим прикладом лінійного представлення є блок-схеми, технічні і архітектурні креслення і т.д. Лінійні структури можуть бути описані аналітично, а з використанням ЕОМ можна порівняно легко формувати складні структури. Ці структури використовують також в тих випадках, коли для вводу зображення застосовують методи слідування за контуром чи іншими траєкторними точками. Яка б структура не використовувалась, в машинному представленні найкраще використовувати єдиний вид структури даних – ланцюговий код. Це послідовність з восьми цифр.

Перевага ланцюгового коду – його компактність. Код дозволяє представити складні дані про зображення, що складається з великого числа областей, у стиснутому вигляді. Оскільки ланцюгові коди представляють дані про границю, їх зручно застосовувати в ЕОМ до задач, що потребують зберігання форми областей. Границі для цілої області можуть зберігатися в формі лінійних списків.

1.1.2 Математичний метод

В основу математичного підходу покладені правила класифікації, які формулюються і виводяться в рамках визначеного математичного формалізму з допомогою принципів загальності властивостей і кластеризації. Коли образи деякого класу представляють собою вектори, компонентами яких є дійсні числа, то цей клас можна розглядати як кластер. Побудова системи розпізнавання, яка базується на реалізації даного принципу, визначається просторовим розміщенням окремих кластерів. Якщо кластери, що відповідають різним класам, рознесені далеко один від одного, то можна користуватися простими схемами розпізнавання, наприклад, класифікацією за принципом мінімальної відстані.

Інший підхід відомий як метод потенціальних функцій, відрізняється тим, що об'єднуються деякі розмиті множини, що описуються так званою потенціальною функцією. При цьому здійснюється апроксимація не розв'язуючою, а дискримінантною функцією. Її значення вираховується шляхом складання значень «потенціалу», який зменшується в міру віддалення від деяких центрів, що вибираються в процесі навчання. Цей підхід є еквівалентним апроксимації дискримінантної функції за допомогою функціонального ряду. Суть полягає в тому, що на просторі вхідних векторів X задається функція, яка називається «потенціалом». Потенціал визначається наближенням двох точок і задається як функція відстані між точками. Потенціальна функція така, що вона монотонно зменшується із збільшенням відстані.

Таким чином, результатом побудови є потенціальне поле, яке розбиває весь простір на дві частини: де значення додатні і від'ємні. Поверхня, на якій потенціал дорівнює нулю, називається розділяючою поверхнею.

1.1.3 Евристичний метод

За основу евристичного підходу взяті інтуїція і досвід людини: в ньому використовуються принципи перерахування членів класу і загальні властивості.

Зазвичай системи, побудовані такими методами, включають набір специфічних процедур, розроблених для конкретних задач розпізнавання. Хоча евристичний підхід відіграє велику роль в розпізнаванні, небагато може бути сказано про загальні принципи синтезу, оскільки розв'язання кожної конкретної задачі потребує використання специфічних прийомів розробки. Це означає, що структура і якість евристичної системи в значній мірі визначається знаннями та досвідом роботи тих, хто її розробляє.

В більшості випадків зображення одного класу зберігають більш-менш постійні розміри і форми. В цих випадках розпізнавання можна здійснити шляхом порівняння зображень із зразками. Такі зразки називають масками, трафаретами, еталонами.

Найпростіший підхід до розпізнавання образів базується на порівнянні їх з еталонами. В цьому випадку деяка множина образів, по одному з кожного класу образів, знаходиться в пам'яті машини. Вхідний образ, що потрібно розпізнати (невідомого класу) порівнюється з еталоном кожного класу. Класифікація базується на раніше вибраному критерії співставлення подібності. Іншими словами, якщо вхідний образ краще відповідає еталону i -го класу образів, ніж в будь-якому іншому еталоні, то вхідний образ класифікується як належність до i -го класу образів. Такий підхід використаний в більшості випадків для читання друкованих літер і банківських чеків. Недоліком цього підходу, тобто співставлення з еталоном, є те, що важко вибрати еталон, який найбільше підходить до кожного класу образів і встановити необхідний критерій відповідності. Ці труднощі особливо суттєві, коли образи, які належать одному класу, можуть мати значні спотворення. Типовим прикладом цього явища є розпізнавання рукописних літер.

Більш вдосконалений підхід ґрунтується на тому, що замість порівняння вхідного образу з еталоном, класифікація базується на деякій множині відібраних вимірів. Ці відібрані виміри називаються ознаками, і є малочутливими щодо звичайних змін і спотворень. В цьому випадку розпізнавання образу можна розглядати в двох задачах.

Перша задача ґрунтується в визначенні, які вимірювання повинні були б бути зроблені на вхідному образі. Звичайний розв’язок задачі про те, що міряти, є певною мірою суб’єктивний, а також залежать від практичних обставин. На сьогодні дуже мало зроблено в конструюванні загальної теорії вибору вимірювальних ознак.

Друга задача розпізнавання образів базується на класифікації (тобто прийнятті розв’язку, чи належить вхідний образ до того чи іншого класу), що має основу на вимірюваннях відібраних ознак.

Одним із перших методів розпізнавання друкованих літер був метод масок. Попередньо центроване зображення літери порівнюється шляхом накладання з масками, трафаретами, виготовленими для всіх літер даного алфавіту. Центрування, тобто перенесення зображення з метою його співставлення з маскою, здійснюється за допомогою фіксації країв знаку, наприклад, верхнього і лівого. Як різновид методу масок можна розглядати метод фрагментів. Аналізуються не всі зображення літери, а тільки деякі “характерні” його ділянки. В залежності від наявності білого або чорного на кожній такій ділянці логічна схема виробляє розв’язок того, яка літера представлена для розпізнавання.

Як бачимо, метод масок і метод фрагментів в їх найпростішому вигляді не застосовується, якщо літера друкується різними шрифтами, або, тим більше, пишеться від руки. Тому були запропоновані гнучкіші методи. Метод проекції полягає в сумуванні (кількості чорного або кількості перетинів вздовж горизонтальних смуг, в яких розміщене зображення літери). Розглядається порядок максимумів і мінімумів цієї суми при переході від смуги до смуги. Аналогічно за допомогою вертикальних смуг аналізується друга проекція тієї ж літери. Отримані дані про чергування максимумів і мінімумів в обох проекціях дозволяють впізнавати літеру незалежно від її розмірів, розташування і деяких змін форми. Цей метод не забезпечує надійного розпізнавання, оскільки існують різні фігури, які дають однакові або дуже близькі проекції, наприклад, косий хрест і контурний ромб з діагоналями, паралельними осям проекції. Крім

того, при деформаціях зображення через випадкові спотворення проекції можуть мати значні зміни. Тому аналіз проекцій сам по собі є важкою задачею розпізнавання.

Великою популярністю користується метод аналізу штрихових елементів знаку. Він полягає в пошуку вертикальних і горизонтальних відрізків, дуг різної орієнтації, а також в вимірюванні їх довжин і в аналізі взаємного розташування. Цей метод, за задумом авторів, повинен забезпечувати розпізнавання літер різних шрифтів.

1.1.4 Розпізнавання простих геометричних фігур і цифр

Значна частина прикладних задач обробки інформації і аналізу даних пов'язана з зображеннями. Цей процес відображає як появу нових технічних засобів отримання інформації, що забезпечують представлення зареєстрованих і накопичених даних у вигляді зображення, так і ріст популярності розпізнавання образів в якості потужної і практичної методології математичної обробки і аналізу інформації і виявлення прихованих закономірностей. В якості прикладів можна відмітити обробку і аналіз даних дистанційного зондування за допомогою фотографування із супутників, об'ємних зображень об'єктів, отриманих голографічними установками, розробку “органів зору” роботів і сучасних медичних систем обстеження і діагностики (рентгенографія, машинна томографія, ендоскопія).

Сучасний етап розвитку обчислювальної техніки відзначається інтенсивним пошуком нових принципів обробки інформації, що диктуються вимогами високої продуктивності систем, а також їх гнучкості і надійності. Вирішувати такі задачі, як цифрова обробка зображень, цифрова фільтрація, розпізнавання образів, обробка метеорологічної та сейсмічної інформації та ін., що пов'язані з обробкою великих масивів інформації в реальному часі, можна використовуючи обчислювальні пристрої з швидкодією на рівні мільярдів операцій за секунду. Такі пристрої повинні відзначатися високою надійністю та універсальністю в класі задач цифрової обробки інформації.

При роботі з зображеннями необхідно вирішити задачі, які виникають в зв'язку з трьома основними проблемами: опис (моделювання) зображення; розробка і вибір математичних засобів обробки і аналізу зображень; апаратна реалізація математичних методів роботи з зображеннями.

Ефективна базова система розпізнавання повинна бути здатна до забирання найкращих характеристик порівнювальних об'єктів, а також проведення масивного спрощення непослідовних моделей. Моделювання об'єктів їхніми локальними геометричними особливостями скористалося перевагою грубої форми і надало можливість швидкої індексації особливостей об'єктів в моделях, що зменшило складність області пошуку перед проведенням зіставлення із взірцем. Ієрархічне об'єктне моделювання ділить контур об'єкта на множину фрагментів, так, щоб кожний фрагмент був множиною особливостей, яка вибирається, як інваріант відносно трансляції і повороту.

Ефективність порівняння залежить від великого степеня операції розпізнавання, яка здатна розпізнати всі контури, також як і фрагменти контурів, зменшуючи область пошуку. Для цього, вибрані особливості повинні бути локальними і достатньо малими для порівняння. Особливості використовуються, як пошукові ключі в деяких швидких схемах індексації.

Проектування систем розпізнавання - достатньо складний ітераційний процес, реалізація якого тісно пов'язана з побудовою уточнюючої математичної або фізико-математичної моделі проектуючої системи. Перша ітерація може бути названа апіорною (вихідною). На її основі складають апіорний алфавіт класів і розробляють апіорний словник ознак. Після вибору алгоритму розпізнавання можна приступити до розв'язування основної задачі, яка складає основу проблеми розпізнавання - визначення алфавітних класів і словника ознак, які забезпечують найбільшу ефективність розв'язків. В теорії розпізнавання найбільш вдалими є розробки з програмування машин для читання машинописних літер або цифр, оскільки вони володіють рядом загальних ознак.

1.2 Лінійне стеження

Нехай ми маємо два суміжні кадри з відеокамери, які являють собою дві бітмапи однакової ширини та висоти. Якщо в полі зору відеокамери були лише нерухомі об'єкти, то (з деякою точністю) ці дві бітмапи (картинки) будуть співпадати по-піксельно.

По-піксельне порівняння картинок є дуже простим алгоритмом, проте воно лише може служити індикатором, чи є рухомі об'єкти в полі зору відеокамери, і не може відповісти на такі прості і важливі питання: як і куди об'єкти рухаються. Якщо ускладнити по-піксельний підхід, і порівнювати один піксель одного кадру з різними пікселями наступного кадру (з різними координатами), то теоретично ми можемо розпізнати зміщення об'єкта на цілу кількість пікселів. Проте якщо об'єкт змістився на дробову кількість пікселів, скажімо на пів-пікселя – то такий ускладнений підхід не спрацює зовсім.

1.1.1 Пояснення принципу на прикладі ідеалізованого зображення

Є другий, аналітичний підхід для обчислення зміщення об'єктів. Як модель, розглянемо спочатку бітмапу, в якій величина пікселя (інтенсивність I) лінійно наростає по координаті абсцис (вісь Ox):

$$I = ax + b \quad (1.1)$$

Якщо на другій бітмапі це саме зображення буде зміщено на величину Δx , то інтенсивність \hat{I} змінюється по закону:

$$\hat{I} = a(x - \Delta x) + b = (ax + b) - a\Delta x \quad (1.2)$$

Маючи ці дві картинки, ми легко можемо обчислити зміщення об'єкта:

$$\Delta x = \frac{\hat{I} - I}{a} \quad (1.3)$$

навіть якщо це зміщення відбулося на дробову кількість пікселів.

Коефіцієнт a в формулі (1.6) є проекцією градієнта інтенсивності на вісь абсцис:

$$(\text{grad}I)_x = I_x = \frac{\partial I}{\partial x} = a \quad (1.4)$$

Очевидно, по формулі $\Delta x = \frac{\hat{I}}{a}$ ми можемо обчислити зміщення тільки вздовж градієнта $I = I(x,y)$, в ортогональному напрямку ми в принципі не можемо помітити переміщення нашого ідеалізованого об'єкта, оскільки при такому переміщенні бітмапа не змінюється. Така невизначеність існує при будь-якій орієнтації градієнта, оскільки ми маємо одне лінійне рівняння відносно двох невідомих переміщень Δx і Δy :

$$I_x \Delta x + I_y \Delta y = I - \hat{I} \quad (1.5)$$

1.2.1 Підготовка зображення для застосування лінійного стеження

Цей метод добре працює для ідеальної модельної картинки, але як бути з реальними зображеннями об'єктів з чіткими контурами? На думку приходить така ідея: ми можемо згладити оригінальне зображення (розфокусувати програмно, усереднюючи сусідні пікселі). В результаті ми одержимо досить гладку функцію інтенсивності, яку методами математичного аналізу можна розкласти в степеневий ряд:

$$I(x + \Delta x) = I_0 + \sum_{i=1}^2 a_i \Delta x_i + \sum_{i,j=1}^2 b_{ij} \Delta x_i \Delta x_j + \dots \quad (1.6)$$

В цій формулі для простоти запису суми, координати (x,y) позначено через $x_1 = x, x_2 = y$.

Розглянемо неперервну модель. Функція інтенсивності $I = I(x,y)$ змінюється неперервно всередині протяжних об'єктів, і скачком змінюється (розриви першого роду) на межі двох об'єктів. Нас цікавить згладжування саме таких скачків. Сума по сусідніх пікселях в неперевній моделі стає інтегралом згортки:

$$\langle I \rangle = \int \phi(x - \xi, y - \eta) I(\xi, \eta) d\xi d\eta \quad (1.7)$$

Якщо функція згладжування ϕ матиме неперервні перші похідні, то і результат згладжування $\langle I \rangle$ буде неперервною функцією і теж матиме перші похідні (скрізь, бо розривів ми позбулись):

$$\frac{\partial \langle I \rangle}{\partial x} = \int \frac{\partial \phi(x - \xi, y - \eta)}{\partial x} I(\xi, \eta) d\xi d\eta \quad (1.8)$$

Функція ϕ , окрім неперервності і диференційовності, повинна бути додатньою в деякому невеликому околі нуля, і дорівнювати нулю за межами цього околу (інакше ми при усередненні можемо домішати пікселі сторонніх об'єктів, і взагалі вийти за рамки картинки). Одним із кандидатів є наступна функція, яку образно можна назвати "біквадратним капелюхом":

$$\phi(x, y) = \begin{cases} \left(1 - \frac{x^2}{a^2} - \frac{y^2}{b^2}\right)^2, & \text{якщо } \frac{x^2}{a^2} + \frac{y^2}{b^2} < 1 \\ 0, & \text{інакше} \end{cases} \quad (1.9)$$

Вона додатня всередині еліпса з півосями a , b , і її перші похідні неперервно перетворюються в нуль на межі. Ясно, що при згладжуванні функцією (1.9) чіткої вертикальної межі ми одержимо приблизно лінійну ділянку на інтервалі, що є часткою півосі a , на більших відстанях ми не можемо користуватися лінійним наближенням. Як показують обчислення, при $\Delta x = a/2$ похибка при обчисленні по формулі (1.3) буде близько 23%.

1.2.2 Оптимальне використання кольору

Нехай наша відеокамера видає кольорову RGB-картинку. Тоді для визначення зміщення Δx ми можемо записати три формули (по одній на кожен колір):

$$R_x \Delta x = -\Delta R; \quad G_x \Delta x = -\Delta G; \quad B_x \Delta x = -\Delta B \quad 1.10$$

Якщо ми розглядаємо згладжену межу між двома кольоровими об'єктами, то очевидно при зміщенні об'єктів усі три формули повинні дати один і той же результат Δx . Проте було б нераціонально обчислювати усі три формули (1.10), проводячи усереднення (1.7) по кожному з трьох кольорів (марнування

процесорного часу). Краще вибрати в ролі інтенсивності деяку лінійну комбінацію кольорів:

$$I = \alpha R + \beta G + \gamma B \quad (1.11)$$

і вже для неї проводити всі усереднення та обчислення. Але як підібрати коефіцієнти α, β, γ найкращим чином? Нехай ми маємо протяжний кольоровий об'єкт, колір якого задається числами $\overrightarrow{Color_1} = (R_1, G_1, B_1)$, на кольоровому ж фоні $\overrightarrow{Color_0} = (R_0, G_0, B_0)$. Кольори можна розглядати як вектори в просторі RGB . Тоді очевидно, найбільш інформативною зміною кольору на межі об'єкта буде проекція вектора різниці:

$$\overrightarrow{DC} = \overrightarrow{Color_1} - \overrightarrow{Color_0} \quad (1.12)$$

В аналітичній геометрії проекція вектора r на вектор a буде відношенням скалярного добутку до довжини вектора a :

$$\text{Pr}_a(r) = \frac{(a \cdot r)}{|a|} \quad (1.13)$$

Повертаючись до формули (1.11), ми можемо по аналогії з формулою (1.13) обчислити коефіцієнти:

$$\alpha = \frac{(R_1 - R_0)}{d} \quad (1.14)$$

$$\beta = \frac{(G_1 - G_0)}{d} \quad (1.14a)$$

$$\gamma = \frac{(B_1 - B_0)}{d} \quad (1.14b)$$

$$d = \sqrt{(R_1 - R_0)^2 + (G_1 - G_0)^2 + (B_1 - B_0)^2} \quad (1.14v)$$

Використання ітерації для підвищення точності лінійного стеження

Розглянемо застосування ітерації на прикладі задачі бінокулярного зору. Нехай ми сфотографували один вертикальний об'єкт (наприклад стовп) двома відеокамерами, які рознесені на деяку відстань по горизонталі. Ясно, що зображення цього стовпа буде зміщене по горизонталі на деяке зміщення Δx при порівнянні двох фотографій. Зміщення по вертикалі не очікується: $\Delta y = 0$,

тому ця задача трохи простіша аніж стеження за рухомими об'єктами. Оскільки нам потрібний тільки градієнт по осі Ox , то функція згладжування (1.9) має бути сплюснутою. Коефіцієнт a повинен щонайменше вдвічі перевищувати сподіване зміщення по іксу: $a \geq 2\Delta x$. На коефіцієнт b не накладено ніяких математичних умов; але якщо ми візьмемо його меншим, то зменшимо кількість обчислень, з другого боку, краще щоб деяке згладжування по осі y відбувалося, щоб позбутися шороховатостей. Тому припустимо, що b дорівнює малому цілому числу (в пікселях), наприклад $b = 1$.

Тепер ми використаємо формули (1.3), (1.4) підставляючи в них згладжену інтенсивність (бажано також з врахуванням кольору по формулі 1.11). Одержимо результат першої ітерації:

$$\Delta_1 x = \frac{I(x) - \hat{I}(x)}{(\text{grad} I)_x} \quad (1.15)$$

Тут ми явно вказали, в яких точках по іксу беруться інтенсивності. Точність цієї формули зменшується зі зростанням зміщення $\Delta_1 x$. Але у випадку великого $\Delta_1 x$, по модулю більшого одиниці, ми можемо провести ще одну ітерацію, змістивши другу картинку $\hat{I}(x)$ на ціле число пікселів що є округленням:

$$n = \text{round}(\Delta_1 x) \quad (1.16)$$

Маємо приблизно:

$$\hat{I}(x + n) \approx \hat{I}(x) + (\text{grad} I)_x n \quad (1.17)$$

а тому наступна формула буде точнішою:

$$\Delta_2 x = n + \frac{I(x) - \hat{I}(x + n)}{(\text{grad} I)_x} \quad (1.18)$$

оскільки другий доданок в цій сумі є малою поправкою.

1.3 Метод Яскравої точки

Існує досить простий алгоритм, оснований на принципах математичної статистики, який дає змогу програмно стежити за яскравими компактними

об'єктами в полі зору відеокамери. Нехай (користуючись іншими міркуваннями та алгоритмами) ми здогадалися, що фрагмент зображення (прямокутник) містить рівномірний (наприклад чорний у випадку нічного неба) фон $I_0 = \text{const}$, на якому видно тільки один яскравий об'єкт: яскраву зірку, місяць, або ліхтар. Причому цей об'єкт цілком знаходиться всередині нашого прямокутника. Реальна інтенсивність I пікселів зображення є функцією від координат:

$$I = I(x, y) \quad (1.19)$$

Ми можемо розглядати відхилення цієї функції від I_0 як величину, пропорційну імовірності знаходження цятки:

$$p = p(x, y) = \frac{I(x, y) - I_0}{\sum_{\xi, \eta} (I(\xi, \eta) - I_0)} \quad (1.20)$$

Сума в знаменнику береться по всіх пікселях нашого прямокутника і характеризує яскравість Y цятки, тобто добуток інтенсивності на площу:

$$Y = \sum_{\xi, \eta} (I(\xi, \eta) - I_0) \quad (1.21)$$

Маючи розподіл імовірності (1.20), ми можемо знайти координати оптичного центру цятки, як математичне сподівання:

$$\bar{x} = \sum_i x_i p_i = \sum_{x, y} x p(x, y) \quad (1.22)$$

$$\bar{y} = \sum_i y_i p_i = \sum_{x, y} y p(x, y) \quad (1.22a)$$

Трохи складніше шукати розміри цятки. Порядок величини розмірів (окремо по горизонталі і окремо по вертикалі) знаходиться через середнє квадратичне відхилення:

$$\sigma_x = \sqrt{(x - \bar{x})^2} = \sqrt{\sum (x - \bar{x})^2 p} \quad (1.23)$$

$$\sigma_y = \sqrt{(y - \bar{y})^2} = \sqrt{\sum (y - \bar{y})^2 p} \quad (1.23a)$$

1.1.1 Розмір круглого диска

Якщо наперед відомо, що яскрава цятка являє собою круглий (або еліптичний) диск на зображенні, то з формул (1.23) ми можемо обчислити радіус цього диска. В неперервному випадку функція щільності імовірності є постійним числом всередині диску:

$$p = p(x, y) = \frac{1}{\pi R^2} \quad \{x^2 + y^2 < R^2\} \quad (1.24)$$

і дорівнює нулю за межами диска. Оскільки в нашому випадку $\bar{x} = 0$, знаходимо дисперсію:

$$D = \sigma^2 = \frac{1}{\pi R^2} \int_{x^2+y^2 < R^2} x^2 dx dy = \frac{1}{\pi R^2} \int_0^{2\pi} \cos^2 \phi d\phi \int_0^R r^3 dr = \frac{R^2}{4} \quad (1.25)$$

звідки

$$R = 2\sigma \quad (1.26)$$

1.3.1 Застосування

Не обов'язково цятка має виділятися білим на чорному – можна і навпаки, темна цятка на світлому фоні, в цьому разі чисельник і знаменник формули (1.20) будуть від'ємними, але імовірність залишиться додатною і ми можемо застосовувати все ті ж формули і з такою ж інтерпретацією. Можна навіть, щоб інтенсивності були приблизно однаковими, а цятка виділялася кольором, як квітка серед трави. Цей алгоритм можна використати також, коли фон I_0 не є однорідним, але може бути якимось чином визначеним як функція $I_0 = I_0(x, y)$ – наприклад фон не змінюється, а яскрава цятка пропливає над ним. В цьому разі ми можемо взяти фон нашого прямокутника із старих кадрів, коли цятка ще не перемістилася в прямокутник.

З огляду на простоту формул (а отже і швидкості реагування), яскраві цятки можуть слугувати прекрасним орієнтиром для робіт в реальній обстановці - дороговказна зірка, маяк, ліхтар, блиск металічного предмета на сонці і т.п. Пошук яскравих цяток має стати обов'язковим для машинного зору.

Особливо треба відмітити, що яскрава цятка (занадто яскрава, наприклад сонце або вогонь електрозварювання) може бути небезпечною для самої відеокамери, і система технічного зору повинна вчасно реагувати: закручувати діафрагму, закриватися шторкою, відвертати погляд відеокамери вбік.

1.4 Виділення об'єкту на зображенні

Для успішного вирішення завдання розпізнавання, необхідно виділити шуканий об'єкт на зображенні, і привести його до нормалізованого вигляду, придатного для розпізнавання.

Основний метод знаходження об'єкту на зображенні – порівняння досліджуваного зображення з еталонним. При кінцевій його реалізації, можливі значні зміни в алгоритмі знаходження об'єктів.

Критерієм успішності виконання алгоритму, можуть служити:

Оцінка успішності людиною.

Аналіз кількості знайдених об'єктів.

Аналіз розмірів знайдених об'єктів.

Аналіз розташування знайдених об'єктів.

Аналіз кольорів знайдених об'єктів

1.1.1 Загальний алгоритм порівняння двох зображень

Визначимо еталонне зображення, як множину I_i точок (x, y) , а досліджуване зображення, як множина J_i . Нехай $c_i(x, y)$ - функція яскравості точки зображень, визначена на множинах I_i и J_i . Один з поширених підходів для визначення областей руху між двома кадрами зображення I_i и J_i ґрунтується на порівнянні відповідних точок цих двох кадрів. [9]. Для цього застосовується процедура формування «різниці» кадрів. Різницею між двома зображеннями, є наступна множина:

$$F(i, j) = \begin{cases} |I(i, j) - J(i, j)| \leq \theta \\ |I(i, j) - J(i, j)| > \theta \end{cases} \quad (1.27)$$

де θ - значення порогового рівня, вибране так, щоб відокремити точки, в яких кадри зображення значно відрізняються один від одного, від точок, в яких яскравість змінилася не набагато, у зв'язку з мерехтінням освітлення, вібрацією камери і іншими перешкодами. Зображення $F(i, j)$ містить передбачувані області руху об'єктів і адитивний шум, що спотворює поточний кадр. Позбавившись від шуму, дозволяють морфологічні операції [10] (операції над бінарним зображенням), такі як ерозія, корозія, а так само різні способи фільтрації і оптимальне (експериментально підібране) значення порогу θ .

Далі, можна обробити знайдені об'єкти, для цього використовуються:

- Метод максимальних площ.
- Метод гістограм.

1.4.1 Метод максимальних площ

Метод приводить знайдені об'єкти до форми, зручнішої для подальшої обробки, і об'єднує окремі області на різницевому зображенні в області за ознакою їх близькості один одному [**Error! Reference source not found.**].

Необхідно виділити області правильної форми, що є достатньо крупними об'єктами. Завдання можна формалізувати таким чином: необхідно розбити безліч крапок F на максимально можливе число непересічних підмножин G_m .

1.4.2 Метод гістограм

У методі використовується гістограма зображення шуканого об'єкту для знаходження об'єкту з такими ж колірними характеристиками на серії зображень.

Потрібно побудувати зображення у відтінках сірого кольору, що містить необхідні нам об'єкти.

Введемо оператора, який перетворить функцію яскравості зображення $G(x, y)$ у функцію кількісного розподілу пікселів з певним значенням яскравості (гістограму) $H(k)$ (де k – чисельне значення яскравості):

$$H(k) = GH \{G(x, y)\}$$

Зворотний оператор GH^{-1} перетворить гістограму в зображення у відтінках сірого.

Алгоритм складається з наступних етапів:

Побудова гістограм шуканого об'єкту h_1 і початкового зображення h_0 ;

Формуємо нову гістограму, як нормований добуток h_1 і h_0 :

$$H(k) = \| h_1 \cdot h_0 \|.$$

Використовуючи зворотне перетворення, отримуємо двовимірну функцію, яка є шуканим зображенням у відтінках сірого:

$$F(x, y) = GH^{-1} \{H(k)\}.$$

1.5 Підготовка зображення до розпізнавання

З погляду завдання розпізнавання, зручніше використовувати зображення об'єктів, що мають однаковий розмір і приблизно однакову орієнтацію в просторі. Проте, алгоритми виділення об'єктів, повертають об'єкти, спотворені перспективою, – різних розмірів і довільно орієнтованих на зображенні.

Для приведення зображення знайденого об'єкту до загального вигляду, необхідно повернути його на потрібний кут. У еталонних і досліджуваних зображеннях об'єктів знаходяться дві контрольні точки, після чого зображення розгортають, так щоб вектора, що сполучають ці точки, збіглися.

Контрольними точками можуть бути, наприклад:

- a) видимий центр зображення;
- b) центр мас зображення;
- c) точка, що помітно відрізняється від останніх за кольором;
- d) центр маркера, поставленого на об'єкті та ін.

Також, необхідно, привести еталонні і досліджувані зображення до одного розміру.

Перераховані вище операції виконуються афінними перетвореннями над матрицями зображень, загальний вигляд яких:

$$\begin{pmatrix} \bar{x}' \\ \bar{y}' \\ 1 \end{pmatrix} = T_{of} \begin{pmatrix} \bar{x} \\ \bar{y} \\ 1 \end{pmatrix}$$

Використовуються окремі випадки афінних перетворень:

Розтягування (стискування) вздовж координатних осей, що задається у вигляді:

$$\begin{aligned} x' &= \alpha x, \alpha > 0, \\ y' &= \beta y, \beta > 0. \end{aligned}$$

Розтягуванню вздовж відповідної осі відповідає значення масштабного множника більшого одиниці. У однорідних координатах матриця розтягування (стискування) має вигляд

$$\mathbf{T}_{af} = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Поворот навколо початкової точки на кут, що описується формулою:

$$\begin{aligned} x' &= x \cos \varphi - y \sin \varphi, \\ y' &= x \sin \varphi + y \cos \varphi. \end{aligned}$$

Матриця обертання (для однорідних координат)

$$\mathbf{T}_{af} = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Перенесення, що задається простими співвідношеннями:

$$\begin{aligned} x' &= x + c, \\ y' &= y + f. \end{aligned}$$

Матриця перенесення має вигляд

$$\mathbf{T}_{af} = \begin{bmatrix} 1 & 0 & c \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix}.$$

1.5.1 Моделювання зовнішнього вигляду об'єкту, використання інструментарію розпізнавання образів (pattern recognition) .

Суть цього підходу полягає в обчисленні деяких числових характеристик зображення модельованого об'єкту (вектора ознак) і застосування різних математичних методів для визначення "схожості" тестових зображень на зображення об'єкту, ґрунтуючись на цих характеристиках.

Наприклад, само зображення необхідного об'єкту можна безпосередньо представити як вектор в багатовимірному просторі і натренувати деякий класифікатор за допомогою набору прикладів зображень об'єктів. Класифікатор в даному випадку означає деякий інструмент, що приймає на вхід зображення, представлене у вигляді вектора в багатовимірному просторі, і видає на виході якусь інформацію, що класифікує вхідне зображення щодо деякої ознаки.

Приклади часто використовуваних класифікаторів:

- ✓ метод найменших квадратів;
- ✓ пряме порівняння по якій-небудь метриці простору векторів ознак (наприклад, сумі різниці кожного елементу вектора) тестового зображення із зображеннями-шаблонами (template-matching);
- ✓ нейромережі (зазвичай для чорно-білих зображень) – на входи нейронної мережі подаються значення елементів вектора, на виходах формується сигнал, що класифікує об'єкт на зображенні;
- ✓ метод опорних векторів (support vector machines) – для розпізнавання зображень;
- ✓ моделювання багатовимірної функції розподілу векторів ознак зображень об'єкту, оцінка вірогідності приналежності тестового зображення до змодельованого розподілу (факторний аналіз, метод головних компонент, аналіз незалежних компонент, лінійний аналіз дискримінанта);
- ✓ моделі, що деформуються;

- ✓ пряме представлення чорно-білого зображення розміру $m \times n$ як вектор породжує простір розмірності $m \times n$ (яскравість кожного пікселя – значення елементу вектора в такому просторі). Тобто зображення порівняно невеликого розширення (100×100) породжує простір розмірності 10,000. Працювати в такому просторі непросто, тому застосовуються різні методики зниження розмірності, наприклад метод головних компонент (principal components analysis, PCA)

Інші приклади характеристик (ознак) зображень, використовуваних для їх класифікації і розпізнавання:

- ✓ статистика розподілу кольорів (у різних уявленнях, зокрема гістограма зображення);
- ✓ статистичні моменти (середнє, дисперсія, приховані Марківські моделі);
- ✓ кількість і властивості графічних примітивів в об'єкті (прямих ліній, кіл – для розпізнавання символів) (на основі перетворення Хафа).

1.6 Сегментація рухомих об'єктів у відео-поточі

Знаходження і відстеження рухомих об'єктів у відеопотоці є важливим завданням для таких застосувань, як охоронні системи, системи аналізу потоку машин на автомагістралі, smart room і так далі. Часто першим кроком до вирішення цих задач є виділення переднього плану. У даній статті буде розглянуто декілька методів віднімання фону (background subtraction), вживаних для вирішення цього завдання. Методи віднімання фону - це методи, які будують модель заднього плану сцени і на кожному кадрі проводять попиксельне порівняння поточного значення в кожному пікселі із моделлю і відповідно до результатів порівняння ухвалюють рішення, належить піксель до заднього плану або до переднього.

Нехай у нас є стаціонарна відеокамера і відеопослідовність, що отримується з неї. Нехай перші n кадрів перед відеокамерою не відбувається

жодного руху, а починаючи з $n+1$ кадру перед камерою можуть появлятися об'єкти, тобто люди машини або щось ще.

Тоді для кожного кадру послідовності потрібно отримати бінарне зображення (маску) переднього плану (рухомих об'єктів). У одержаній масці 0/чорний відповідає задньому плану, а 1/білий відповідає передньому плану. Приклад успішної сегментації представлено на Рис. 1.1.

Варто додатково відмітити, що вимога про те, що перші n кадрів перед камерою не відбувається жодного руху не завжди може бути виконана. Як приклад можна розглянути відеопослідовність із виглядом на вулицю, на якій знаходиться дерево із листям, що коливається від вітру. Подібне явище ми називатимемо динамічним заднім планом. Дану вимогу можна замінити на те, що перші n кадрів перед камерою не з'являється ніяких об'єктів інтересу тобто видно тільки лише задній план, статичний або динамічний.

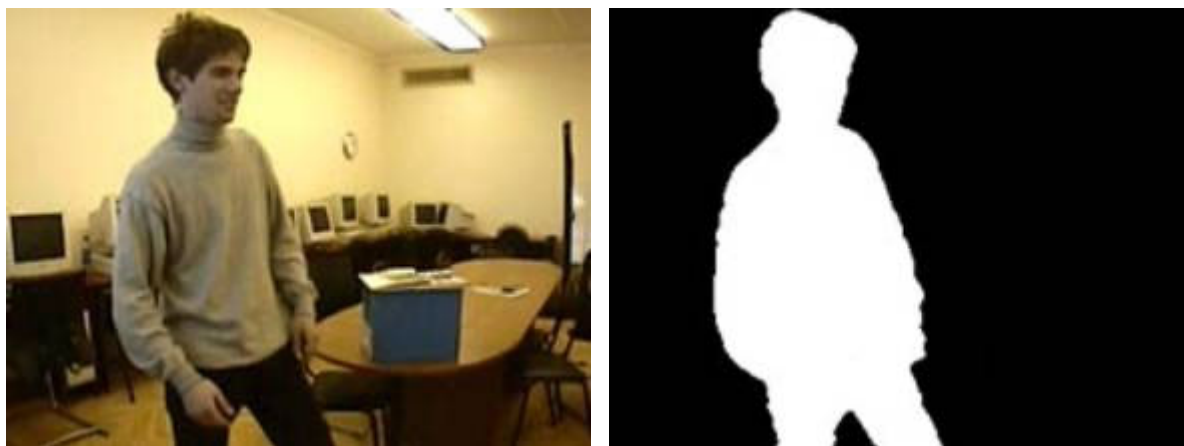


Рис. 1.1. Приклад сегментації кадру відеопослідовності

1.6.1 Простий алгоритм і його проблеми

Розглянемо простий алгоритм віднімання фону і на його прикладі познайомимося із основними проблемами, з якими стикаються такого роду алгоритми. Вважатимемо, що з відеокамери поступають півтонові зображення (якщо з камери надходять кольорові зображення, то їх очевидно можна привести до півтонових, наприклад використовуючи формулу $I = 0,3 \cdot R + 0,59 \cdot G + 0,11 \cdot B$). Алгоритм зберігає перший кадр відео-

послідовності, а потім для кожного наступного кадру застосовує поріг δ до модуля різниці поточного і збереженого зображення по кожному пікселю.

Отже:

- 1) Зберігаємо перше зображення відеопослідовності. Позначимо його B .
- 2) Для кожного наступного такого кадру (позначимо поточний кадр C) якщо $|B[i][j] - C[i][j]| \leq \delta$, $i = 1..w$, $j = 1..h$, де w і h це ширина і висота картинки відповідно, то піксель $[i][j]$ вважається передньоплановим, інакше він вважається задньоплановим.

- 3) Для усунення шуму до отриманого бінарного зображення можна застосувати медіанний фільтр або операції математичної морфології.

Відзначимо 2 види помилок такого алгоритму: передньоплановий піксель був класифікований як задньоплановий (наприклад в наслідку шуму або через те, що колір передньопланового об'єкту збігається із кольором заднього плану) і задньоплановий піксель був класифікований як передньоплановий (наприклад із-за шуму). Змінюючи параметр δ (поріг) і параметри подальшої наступної фільтрації можна регулювати чутливість алгоритму тобто рівні цих помилок. Безперечними достоїнствами даного алгоритму є виняткова простота реалізації і висока продуктивність. Не дивлячись на ці достоїнства існує ряд проблем, які роблять цей алгоритм практично непридатним на практиці.

1.6.2 Зміни заднього плану

Уявимо собі наступну, цілком реальну, ситуацію: відеокамера установлена на вулиці і в полі її видимості знаходиться машина, через деякий час приходить хазяїн машини сідає в неї і виїжджає. Подібна ситуація є великою проблемою для викладеного алгоритму віднімання фону. Річ у тому, що після того, як машина поїде, камері відкриється частина сцени, яку машина собою загороджувала. Ця частина сцени в загальному спільному випадку матиме колір відмінний від кольору машини і, таким чином, сприйматиметься алгоритмом як передній план.



Рис. 1.2. Проблема зміни заднього плану. Зверху перший кадр послідовності. Звернете увагу на коробку на столі. Знизу кадр і результати сегментації, після того, як коробку із столу прибрали

Якщо в задньому плані сцени подібного роду зміни відбуватимуться досить часто, то у результаті велика частка сцени сегментуватиметься в передній план і дані сегментації будуть абсолютно непридатні для аналізу і обробки. Ілюстрацію даної проблеми можна побачити на Рис. 1.2.

1.6.3 Зміна освітлення

Зміни освітлення практично повністю змінюють колірні характеристики сцени. У результаті із-за сильної зміни освітлення алгоритм сегментуватиме в передній план велику частину сцени, що зрозуміло цілком неприйнятно. Варто розбити зміни освітлення на 2 групи: швидкі (такі, наприклад, як включення

світла в кімнаті) і повільні (наприклад зміна освітлення в течії дня). З повільними і швидкими змінами освітлення можна боротися по-різному.

1.6.4 Рухомі тіні

Знайти тінь на зображенні або у відеопотоці для людини не складає жодних труднощів тоді як автоматичне знаходження тіней - досить складне завдання. Алгоритм віднімання фону стикається із проблемою рухомих тіней тобто тіней від рухомих об'єктів сцени. Річ у тому, що падіння тіні на об'єкт досить сильно міняє колір об'єкту, і саме тому викладений вище алгоритм класифікує тіні як передній план. Ілюстрацію проблеми рухомих тіней можна побачити на Рис. 1.3.



Рис. 1.3. Проблема рухомих тіней. На рисунку виразно видно <хмару> під людиною.

1.6.5 Динамічний задній план

Викладений алгоритм не в змозі також впоратися з проблемою динамічного заднього плану. Запам'ятавши перший кадр відеопослідовності алгоритм надалі сегментуватиме такі <нецікаві> рухи, як наприклад рух листя на дереві або рух води.

1.6.6 Шум камери

Різні камери шумлять по-різному і це також може бути проблемою для даного алгоритму віднімання фону. Неадаптивний поріг доводиться

настроювати вручну для різних камер і різних налаштувань камери, що зрозуміло небажано для автоматичних систем.

1.6.7 Алгоритми, засновані на імовірнісних моделях

Вище перелічені проблеми роблять алгоритм із неадаптивним порогом практично неприйнятним на практиці і тепер самий час перейти до складніших алгоритмів, які ці проблеми вирішують.

1.6.8 Нормальний розподіл

Почнемо з алгоритму, що використовує нормальний розподіл. Знову вважаємо, що з камери надходять півтонові зображення. Даний алгоритм моделює кожен піксель заднього плану за допомогою одновимірної нормально розподіленої величини, тобто:

$$B \sim N(x, \mu, \delta^2)$$

Перші n кадрів (коли камері видно тільки задній план) для кожного пікселя алгоритм проводить оцінку параметрів моделі тобто μ і δ^2 використовуючи апарат математичної статистики а потім, після того, як навчання закінчене, попіксельно використовує імовірнісний поріг для того, щоб визначити чи належить значення пікселя задньому плану або передньому. Отже:

1) Навчання моделі заднього плану по перших n кадрах. Для оцінки параметрів моделі для кожного пікселя можна застосовувати вибіркове середнє і середнє квадратичне відхилення значень в даному пікселі тобто $\mu = \frac{1}{n} \cdot \sum_{i=1}^n X_i$,

$$\delta^2 = \frac{1}{n-1} \cdot \sum_{i=1}^n (X_i - \mu)^2 \text{ де } X_1 \dots X_i - \text{значення в пікселі по перших } n \text{ кадрах.}$$

2) Далі для кожного наступного кадру і для кожного пікселя. Позначимо за C значення в поточному кадрі для (i, j) пікселя. Якщо $\frac{|\mu - c|}{\delta} \leq e$ то даний піксель сегментується в задній план, інакше він сегментується в передній план.

3) Для усунення шуму до отриманого бінарного зображення можна застосувати медіанний фільтр або операції мат. морфології

Параметр e впливає на чутливість алгоритму і вибирається із наступних міркувань. Фіксується деяке число $a \in (0,1)$. Після цього відносно e розв'язується рівняння $P\left(\frac{\mu - \xi}{\delta} \leq e\right) = a$, де $\xi \sim N(\mu, \delta^2)$. Число a є не що інше, як вірогідність того, що відхилення викликане шумом камери попадатиме у вказану вище окружність. Дуже часто користуються $e = 3$ (т.з. Mahalanobis distance) якому відповідає значення $a \approx 0,98$. Даний алгоритм вирішує проблему шуму камери оскільки має адаптивний імовірнісний поріг, але решта всіх проблем залишається в силі. Відмітимо, що даний алгоритм допускає реалізації не тільки для півтонових але і, наприклад, для RGB зображень, що покращує якість роботи алгоритму. При реалізації в RGB слід використовувати 3-х мірний нормальний розподіл замість одновимірною і побудувати імовірнісний поріг для 3-х мірної величини.

1.6.9 Суміш нормальних розподілів

Алгоритм створює попиксельну модель сцени використовуючи суміш нормальних розподілів і з надходженням кожного нового кадру оновлює модель і класифікує кожен піксель, що належить до заднього або до переднього плану. Таким чином:

$$S \sim \sum_{p=1}^k w_p \cdot N(x, \mu_p, \delta_p^2),$$

$$\text{де } N(x, \mu, \delta^2) = \frac{1}{\sqrt{2\pi\delta}} e^{-\frac{(x-\mu)^2}{2\delta^2}}.$$

Кожному доданку в сумі відповідає процес в пікселі сцени, який характеризується параметрами нормального розподілу (мат. очікуванням і дисперсією) і коефіцієнтом w , який називається вагою і є показником того, наскільки часто даний процес в даному пікселі потрапляв у поле зору камери. Параметр (максимальна кількість Гаусіанів) вибирається відповідно до ресурсів

комп'ютера, зазвичай беруть значення від 3 до 5. Використовуючи і навчаючи таку модель можна створити алгоритм, який справляється практично із всіма перерахованими вище проблемами (окрім рухомих тіней). Пропонований алгоритм є модифікацією стандартного алгоритму суміші нормальних розподілів [1]. Знову вважатимемо, що з камери поступають півтонові зображення. Отже:

1) На першому кадрі відеопослідовності відбувається ініціалізація моделі. У кожному пікселі створюється один процес із наступними параметрами: $w = 1$, $\mu = c$, $\delta^2 = \delta_{std}^2$, де c - поточне значення в даному пікселі а δ_{std}^2 - дисперсія за замовчуванням (вибирається вручну). Всі пікселі сегментуються в задній план.

2) Для кожного наступного кадру і для кожного пікселя:

1. Пошук процесу, який задовольняє значення даного пікселя c . Для кожного процесу в моделі застосовується поріг $\frac{|\mu - c|}{\delta} \leq e$. Якщо поточне значення c для деякого процесу задовольняє поріг, то даний процес позначається як поточний, і переходимо до пункту 3. Якщо поріг не виконався ні для одного процесу то переходимо до пункту 2.

2. Створення нового процесу. Оцінки мат. очікування і дисперсії вибираються наступним чином: $\mu = c$, $\delta^2 = \delta_{std}^2$. Якщо кількість процесів в моделі вже рівна k , то шукається процес із найменшою вагою, його вага не міняється, а решта параметрів прирівнюється до параметрів нового процесу, даний процес позначається як поточний і переходимо до пункту 5. Якщо кількість процесів в моделі ще не досягла максимуму то новий процес додається до списку процесів, його вага прирівнюється 0, цей процес позначається як поточний і переходимо до пункту 4.

3. Оновлення статистики поточного процесу. Оцінки мат. очікування і дисперсії оновлюються за допомогою низькочастотного фільтру рекурсивного згладжування. Позначимо за μ_{t-1} і δ_{t-1}^2 оцінки параметрів поточного процесу на попередньому кроці, а за μ_t і δ_t^2 оцінки на поточному кроці, тоді

$\mu_t = (1 - \alpha_1) \cdot \mu_{t-1} + \alpha \cdot c$, $\delta_t^2 = (1 - \alpha_2) \cdot \delta_{t-1}^2 + \alpha_2 (c - \mu_t)^2$, де α_1, α_2 - параметри фільтру дозволяють регулювати швидкість навчання.

4. Оновлення ваг процесів. Позначимо за $w_{i,t-1}$ вагу i -того процесу на попередньому кроці, а за $w_{i,t}$ вагу i -того процесу на поточному кроці, тоді $w_{i,t} = (1 - \alpha_3) \cdot w_{i,t-1} + \alpha_3 \cdot M(i)$, де α_3 - параметр відповідає за швидкість зміни ваги, а $M(i)$ - функція рівна 1 при i рівному індексу поточного процесу і рівна 0 при решті всіх значень i .

5. Класифікація пікселя. Алгоритм класифікації дуже простий: застосування порогу до ваги поточного процесу. Якщо значення ваги поточного процесу більше ніж значення порогу, то піксель класифікується як задньоплановий, інакше він класифікується як передньоплановий. Поріг слід вибирати з інтервалу $(0, 1)$.

3) Для усунення шуму до отриманого бінарного зображення можна застосувати медіанний фільтр або операції мат. Морфології.

Метод, використовуваний для класифікації пікселів очевидно допускає приналежність до заднього плану відразу декількох процесів. Саме після цього за наявності динамічного заднього плану алгоритм через деякий час адаптується до його рухів, що повторюються. Більше того, алгоритм здатний адаптуватися до таких рухів сцени як дощ або сніг. Також алгоритм здатний адаптуватися до змін заднього плану, оскільки через деякий час після зміни, вага створеного при зміні процесу перевищить поріг і він почне сегментуватися в задній план. До повільних змін освітлення алгоритм здатний адаптуватися завдяки системі оцінки параметрів, оскільки у міру плавної зміни кольору пікселів заднього плану алгоритм переучуватиметься на нові значення. До швидких змін освітлення алгоритм адаптується також, як він адаптується до змін заднього плану.

1.6.10 Виявлення тіней

Точне виявлення і придушення рухомих тіней являється однією з головних проблем алгоритмів віднімання фону оскільки кожен рухомий об'єкт відкидає тінь. Це може привести до серйозних спотворень форми об'єктів, а також може призводити до того, що різні об'єкти зіллються в один, що може сильно ускладнити подальший аналіз і обробку.

Алгоритм тінеподавлення, що використовує локальні (попиксельні) властивості тіней представлений нижче.

Розглянемо деяку точку сцени і нехай вона має колір (R, G, B) , тоді після падіння на цю точку тіні вона матиме колір $\alpha \cdot (R, G, B)$, де коефіцієнт α визначає те, наскільки впала освітленість в точці. Виявляється, що цього факту цілком достатньо, аби побудувати непогану систему придушення тіней. Алгоритм тінеподавлення представлено для RGB реалізації алгоритму з одним нормальним розподілом, та за аналогією дану систему тінеподавлення можна інтегрувати в алгоритм з порогом і алгоритм з сумішшю нормальних розподілів. Нехай (R_1, G_1, B_1) – колір заднього плану в деякому пікселі (тобто оцінка мат. очікування 3-х мірної випадкової величини, що моделює задній план в даному пікселі) а (R_2, G_2, B_2) – поточне значення в цьому пікселі. Спершу перейдемо в колірний простір YUV тобто $(R_i, G_i, B_i) \rightarrow (Y_i, U_i, V_i)$ $i = 1, 2$. У YUV компонента Y визначає інтенсивність а компоненти U, V визначають колірність. Таким чином при падінні тіні на об'єкт значення Y повинне суттєво зменшитися, а значення U, V практично не поміняються. Таким чином значення в пікселі класифікується як тінь, якщо $|U_1 - U_2| \leq \alpha$, $|V_1 - V_2| \leq \beta$, $Y_2 < Y_1$, де α, β – значення порогів визначаючи чутливість системи тінеподавлення. На Рис. 1.4 показаний приклад роботи системи виявлення тіней для RGB реалізації алгоритму із сумішшю нормальних розподілів. Варто відмітити, що це не єдиний метод придушення тіней, який використовує їх локальні властивості. У [3] можна прочитати про схожу систему, яка використовує HSV замість YUV .



Рис. 1.4. Придушення тіней. Тіні відмічені сірим кольором.

2. ПРАКТИЧНА ЧАСТИНА

2.1 Алгоритм стабілізації образу

Використовуючи як засоби реєстрування відеофайлів нестационарні відеокамери чи фотоапарати, завжди постає проблема стабілізації образу. Щоб якомога точніше співставити два послідовні кадри ми вирізаємо два прямокутники із одного зображення більший шириною і висотою відповідно r_1 і s_1 і другий, менший розмірами r_2 і s_2 , як показано на Рис. 2.1. Після цього переміщаючи менше зображення по більшому шукаємо суму модулів різниці між зображеннями. Після того як ми пересунемо по піксельно менше зображення на z_w пікселів у право і ліво, і z_h пікселів у гору і низ, ми визначаємо найменше значення (2.2) модуля різниці двох зображень (2.1) і знаходимо координати зміщення.

$$SUM[z_w, z_h] = \sum_{i,j} C[i, j]. \quad (2.1)$$

$$[рядок, стовпець, мінімальне значення] = \min(SUM) \quad (2.2)$$

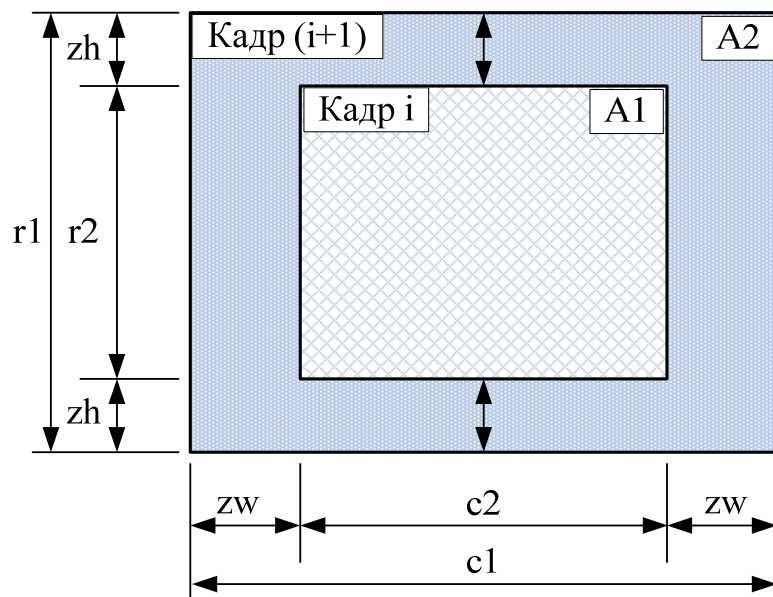


Рис. 2.1. Схема розміщення

Блок-схема алгоритму стабілізації кадрів представлена на Рис. 2.2. Як бачимо з блок схеми, кількість ітерацій на пряму залежить від параметрів заступу zh і zw . Тому коли зображення відносно стабільне то значення zh і zw рекомендовано вибирати в межах $5 \leq zh \leq 10$, $5 \leq zw \leq 10$.

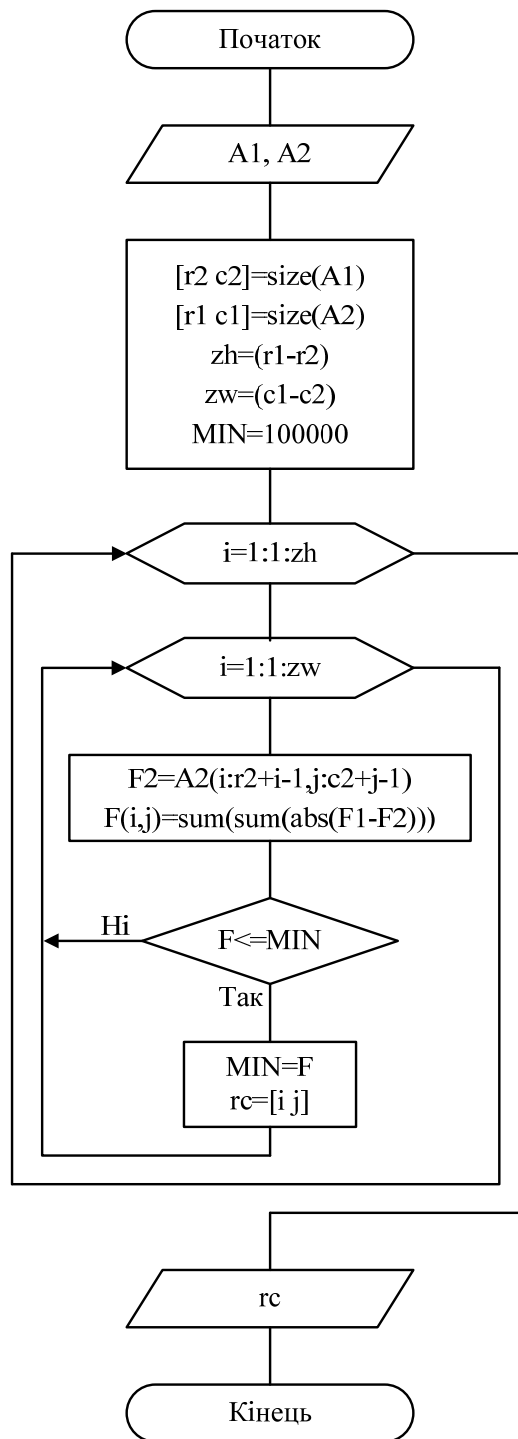


Рис. 2.2. Блок-схема алгоритму стабілізації зображення

На Рис. 2.3(а) і Рис. 2.3(б) представлена відповідно різниця двох послідовних кадрів без стабілізації і з використанням стабілізації образу.

а)



б)



Рис. 2.3. Різниця двох послідовних кадрів: а) без стабілізації образу,
б) з застосування стабілізації образу



Рис. 2.4. Різниця двох послідовних кадрів без стабілізації образу



Рис. 2.5. Різниця двох послідовних кадрів з використанням стабілізації образу

2.2 Алгоритм знаходження фону зображення

Одним з важливих етапів реалізації системи визначення інтенсивностей транспортних потоків є визначення фону зображення без рухомих об'єктів, якими в даному випадку є транспортні засоби.

Проаналізувавши відомі методи знаходження фону зображення, було розроблено алгоритм який базується на простому алгоритмі визначення фону [23, 24]. Алгоритм зберігає перший кадр відеопослідовності, а потім для кожного наступного кадру застосовує поріг δ до модуля різниці поточного і збереженого зображення по кожному пікселю. Безперечними достоїнствами даного алгоритму є виняткова простота реалізації і висока продуктивність.

Основними проблемами, з якими стикаються такого роду алгоритми є:

1) передньоплановий піксель був класифікований як задньоплановий (наприклад в наслідок шуму або через те, що колір переднепланового об'єкту збігається із кольором заднього плану) і задньоплановий піксель був класифікований як переднеплановий (наприклад із-за шуму).

2) фон зображення змінився, наприклад запаркований автомобіль почав рух.

Провівши дослідження з різними сценами було встановлено, що при низькому значенні порогу потрібно провести велику кількість ітерацій для знаходження фону зображення що приводить до зростання часу обчислень, а в деяких випадках взагалі не вдається повністю визначити фон. У випадках, коли значення порогу велике, кількість ітерацій зменшується але втрачається якість фонового зображення, і на фоні проявляються обриси рухомих об'єктів. Для вирішення проблем з якими стикаються даного роду алгоритми було запропоновано використовувати не одне порогове значення для класифікації пікселя до заднього або переднього плану, а три порогові значення. Для цього було розроблено алгоритм як на Рис. 2.6, особливостями якого є:

1) при умові $|R1[i, j] - R1[i, j]| \leq 5$, $i = 1..w$, $j = 1..h$, де w і h це ширина і висота картинки, піксель $R2[i, j]$ в будь-якому випадку присвоюється фоновому зображенню $BG[i, j] = R2[i, j]$, це призводить до того, що навіть якщо на

попередньому кроці було визначено піксель який задовольняв умову $|C[i, j] - C[i - 1, j - 1]| \leq 20$, то він заміниться більш точним;

2) для зменшення обчислювальної складності проводиться перевірка поточного фонового зображення, якщо $BG[i, j] = 0$, то проводяться подальші перевірки, а в разі якщо $BG[i, j] \neq 0$, то переходимо до наступного пікселя;

3) експериментально встановлено що хороші результати досягаються для другого порогового значення рівного 20, при цьому не суттєво втрачаємо на якості, але при середніх інтенсивностях руху для м. Львова кількість ітерацій скорочується до 5;

4) встановлено, що при низькій якості відео близько 10 % пікселів не задовольняють другого порогового значення, навіть при суттєвому збільшенню кількості ітерацій. Тому, якщо встановлено що це останній кадр то невизначеним пікселям фонового зображення присвоюється середнє значення пікселів з поточного і попереднього кадру, для того щоб згладити значення пікселя у випадку, якщо він є рухомих об'єктом.

Для дослідження розробленого алгоритму проведено моделювання для різних значень порогу. Результати представлені у вигляді рисунків, для наглядного їх порівняння. На Рис. 2.7, Рис. 2.8 представлено відповідно визначені фони за допомогою алгоритму з одним пороговим значенням, які становлять відповідно 10 і 20. А на Рис. 2.9 представлено фонове зображення отримане за допомогою розробленого алгоритму (Рис. 2.6).



Рис. 2.7. Фонове зображення при пороговому значенні 10



Рис. 2.8. Фонове зображення при пороговому значенні 20



Рис. 2.9. Фонове зображення одержане при використанні розробленого алгоритму

На Рис. 2.10 - Рис. 2.14 зображено від першої по п'яту ітерації знаходження фону зображення за допомогою розробленого алгоритму. Як видно з рисунків, 5 ітерацій для даної ділянки дороги цілком достатньо, очевидно що у випадках з більшою інтенсивністю руху, кількість ітерацій доведеться збільшити.



Рис. 2.10. Перша ітерація знаходження фону зображення



Рис. 2.11. Друга ітерація знаходження фону зображення



Рис. 2.12. Третя ітерація знаходження фону зображення



Рис. 2.13. Четверта ітерація знаходження фону зображення



Рис. 2.14. П'ята ітерація знаходження фону зображення

2.3 Трекінг (супроводження) об'єктів

Метою трекінгу є встановлення відповідності між об'єктами або їх частинами в послідовності кадрів, а також визначення їх траєкторій і швидкості руху. Особливу складність цьому завданню додають зміни ракурсів об'єктів під час руху, а також їх часткове або повне перекриття, коли один об'єкт повністю загороджує інший. В більшості літературних джерел **[Error! Reference source not found.]** встановлення відповідності між об'єктами відбувається за допомогою кореляційного аналізу.

Недоліками таких алгоритмів є:

- 1) складність обчислень;
- 2) проблема виявити об'єкт (автомобіль), коли його зступає інший;

У [42] виявлення транспортного засобу здійснюється визначаючи різні особливості характеристик на чорно-білому зображенні. Процес виявлення використовує тінь і геометричні особливості транспортного засобу, щоб стверджувати чи це є транспортний засіб.

У [43, 44] використовувалися нейронні мережі для виявлення транспортного засобу. Нейронні мережі мають один недолік, а саме те, що немає гарантії, що вони досягають глобального мінімуму (в даному випадку немає вирішень закритої форми для виявлення транспортного засобу).

У [44] процес виявлення залежить від значення світлової інтенсивності. Коли значення світлової інтенсивності різко зменшується, то використовується певний інтервал невизначеності, який повинен використовуватися, щоб вирішити, чи це транспортний засіб, чи ні. Коли значення інтенсивності більше, ніж цей інтервал, то це є транспортний засіб і коли інтенсивність є меншою ніж цей інтервал транспортний засіб не виявляється. Цей підхід реагує на зміни середовища.

Проаналізувавши дані методи, поставлена задача розробити власний алгоритм, у зв'язку з тим, що при визначенні інтенсивності руху ми повинні встановлювати зв'язок між багатьма об'єктами, що є ресурсоемним.

Саме тому поставлена задача розробити алгоритм який був би простіший за складністю і потребував менше комп'ютерних ресурсів для обробки даних, але не менш ефективним.

Розроблений алгоритм полягає в наступному:

На першому кадрі ми рахуємо всі об'єкти, і визначаємо центри мас цих об'єктів. На наступному кадрі знову визначаємо центри мас всіх об'єктів і порівнюємо значення кожного центру мас на поточному кадрі з центрами мас об'єктів з попереднього кадру. Якщо найменша відстань між центрами мас є меншою за ширину об'єкта, тобто $l_1 < l_{car}$ як на Рис. 2.15, то значення лічильника залишається таке саме. Якщо $l_1 > l_{car}$ – це означає, що на попередньому кадрі цього об'єкта не було, відповідно значення лічильника збільшується на 1.

Основними перевагами розробленого алгоритму є:

- 1) вирішується проблема коли автомобіль буде заступатися іншим автомобілем;
- 2) простота обчислень;

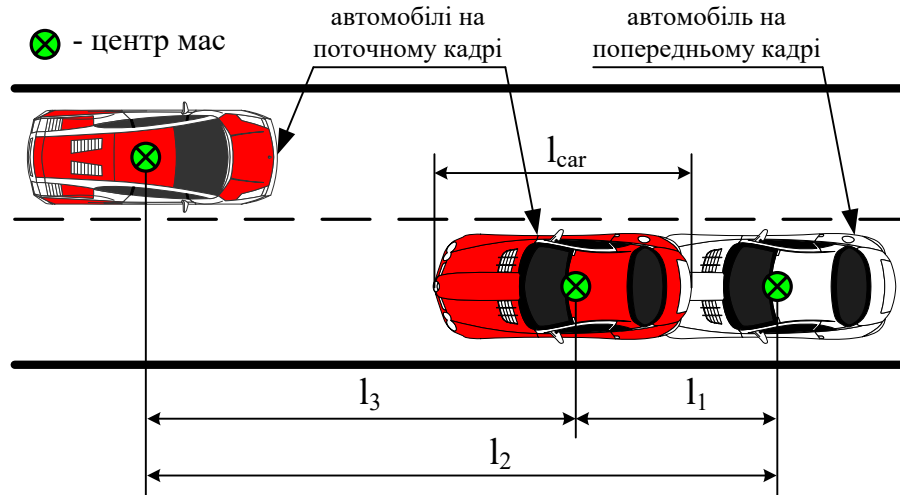


Рис. 2.15. Зображення автомобілів на двох послідовних кадрах

Для того, щоб визначити чи поточний центр мас належить якомусь з об'єктів на попередньому кадрі було розв'язано наступну задачу.

Задача полягала в наступному: якщо ми маємо на кадрі автомобіль з шириною в середньому 4м., в один кадр по ширині поміщається 3,5 автомобіля. То ширина кадру з роздільною здатністю 640 пікселів вміщає:

$$4(авт.) \cdot 3,5(м) = 14(м)$$

Наступним кроком визначаємо скільки пікселів припадає на 1(м):

$$\frac{640(пікселів)}{14(м)} \approx 46(пікселів)$$

Відповідно ширина автомобіля в пікселях буде становити:

$$46(пікселя) \cdot 4(м) = 183(пікселя)$$

Якщо у відеопотоці 30 кадрів на секунду, тоді період $t_{між кадрами}$ між двома аналізованими кадрами буде становити:

$$t_{між кадрами} = \frac{кількість кадрів}{крок}$$

де *кількість кадрів* – кількість кадрів за секунду у відео потоці;

крок – крок в кадрах з яким відбувається порівняння двох кадрів.

а) якщо ми будемо розглядати кожний кадр: $1/30$ (с).

б) якщо ми будемо розглядати кожний третій кадр: $3/30$ (с).

в) якщо ми будемо розглядати кожний п'ятий кадр: $5/30$ (с).

г) якщо ми будемо розглядати кожний сьомий кадр: $7/30$ (с).

Для того щоб можна було визначити з яким кроком потрібно проводити аналіз відео, не втрачаючи на точності результатів побудовано два графіки. На першому Рис. 2.16. представлено на яку кількість пікселів зміститься автомобіль в наступному кадрі з кроком кадрів 1, 3, 5 і 7 відповідно, при швидкостів від 5 до 80 (км/год) і масштабі 15(м) до 640 (пікселів) по ширині. Рис. 2.17. представляє ті ж самі залежності, тільки при масштабі 30 (м) до 640 (пікселів) по ширині.

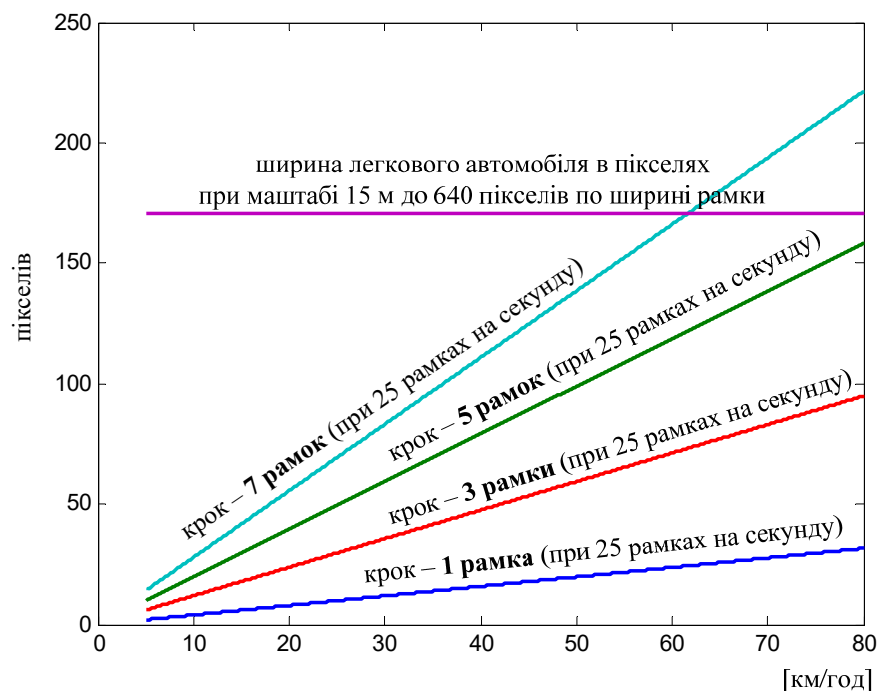


Рис. 2.16. Графік залежності переміщення автомобіля від швидкості

Провівши даний аналіз, встановлено, що алгоритм буде ефективно працювати навіть з кроком 5 кадрів, що суттєво зменшить час обчислень. Тому, що навіть при швидкості 80 (км/год), через 5 кадрів центр маси автомобіля не

змістяться більше як на ширину автомобіля, що виключає варіант, що в цьому місці може бути інший автомобіль.

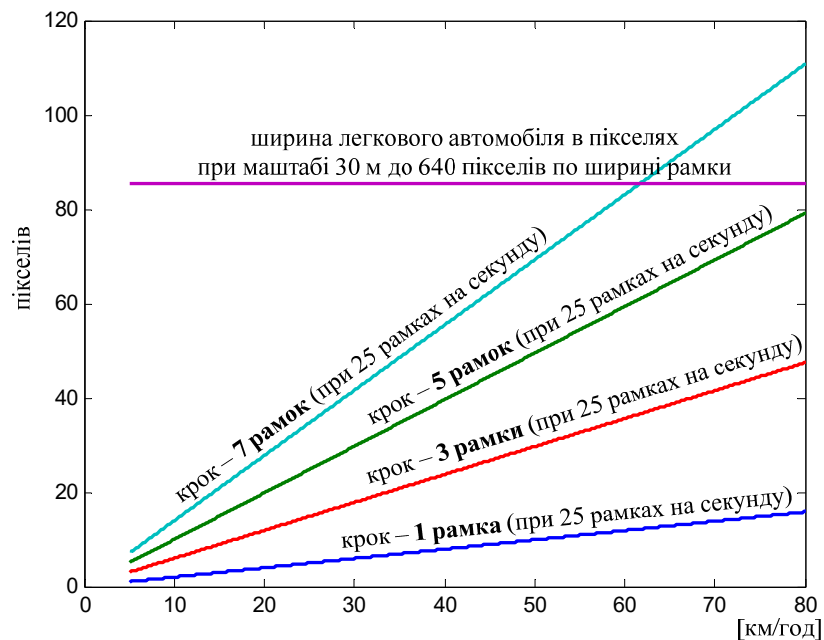


Рис. 2.17. Графік залежності переміщення автомобіля від швидкості

2.4 Особливості реалізації в системі MatLab

Загальний алгоритм роботи програми визначення інтенсивності транспортних потоків представлений на Рис. 2.18. Де у першому блоці зчитування послідовності кадрів з відеофайлу відбувалося за допомогою функції AVIREAD, яка дозволяє зчитувати avi-файли. Дана функція також дозволяє зчитувати не всі кадри, а з певним кроком, що суттєво зменшує затрати часу, так як для визначення інтенсивності транспортних засобів достатньо зчитувати кожний третій кадр.

Для перетворення кольорового зображення у півтонове використовувалася функція *rgb2gray*, яка дозволила зменшити розмірність кольорового зображення $[h, w, 3, nr]$ на чорнобіле $[h, w, 1, nr]$,

де h - кількість пікселів по висоті;

w - кількість пікселів по ширині;

nr - кількість кадрів.

Наступним кроком півтонове зображення розмивалося медіанним фільтром [Error! Reference source not found., Error! Reference source not

found.]. У системі MatLab використовувалася вбудована функція – `medfilt2` для зменшення шумів. Медіанна фільтрація є окремим випадком рангової фільтрації. Функція `D=medfilt2(S [m n])` створює півтонове зображення `D`, кожен піксель якого формується таким чином. Пікселі початкового півтонового зображення `S`, відповідні всім елементам маски фільтру розміру `mn`, складають впорядковану послідовність `A`. Пікселю `D(r, c)`, де `r` і `c` – координати поточного положення центрального елемента маски, привласнюється значення медіани послідовності `A`. Операція застосовується не рекурсивно для всіх положень маски.

Медіаною впорядкованої послідовності `A(i)`, де $i=1...N$, називається величина $A((N + 1)/2)$, якщо N - непарне, і $(A(N/2)+A((N+2)/2))/2$, якщо N - парне.

Для того, щоб розміри зображень `S` і `D` були однаковими, при проведенні обчислень зображення `S` тимчасово доповнюється необхідною кількістю рядків і стовпців нульових пікселів. Формат представлення даних результуючого зображення `D` збігається з форматом початкового зображення `S`.

Якщо вектор `[m n]` при виклику функції `D=medfilt2(S)` не заданий, то як маска фільтру використовується маска розміру 3×3 .

Стабілізація образу відбувалася за розробленим алгоритмом, який описаний у попередньому розділі, блок-схема якого представлена на Рис. 2.2.

У четвертому блоці блок-схеми перед створення маски об'єкту використовувалися операції математичної морфології. При цьому використовувалися такі функції як:

bwareaopen – видалення малих об'єктів. Функція `BW2=bwareaopen(BW,P)` видаляє з бінарного зображення всі зв'язні компоненти (об'єкти), площа яких менше `P` пікселів, і поміщає результат обробки в зображення `BW2`. За замовчуванням зв'язаними є 8 елементів для двохвимірного зображення, 26 – для трьохвимірного і `conndef(ndims'(BW),'maximal')` – для багатовимірних. У функції `BW2=bwareaopen(BW,P,CONN)` додатково вказується параметр зв'язності.

Вимоги до початкових даних: початкове зображення повинно бути представлено логічним або числовим нерозрідженим масивом. Результат обробки, який поміщається в зображення BW2, є логічним масивом.

imfill – заповнення регіонів зображення; Функція $BW2 = \text{imfill}(BW, \text{LOCATIONS})$ виконує операцію заливки фоновими пікселями початкового зображення BW, починаючи з точок, визначених параметром LOCATIONS. Параметр LOCATIONS має бути вектором розмірності P1, в якому містяться лінійні індекси початкових точок. Параметр LOCATIONS може бути також матрицею $\text{Pndims}(BW)$, в якій містяться масиви індексів початкових елементів. Функція $BW2 = \text{imfill}(BW, 'holes')$ заповнює отвори на початковому бінарному зображенні.

strel – створення морфологічної структури елементів.

imerode – Функція $IM2 = \text{imerode}(IM, SE)$ проводить операцію потоншення півтонових, бінарних або упакованих бінарних зображень IM, повертаючи витончене зображення IM2. Аргумент SE є структурним елементом об'єкту або масивом структурних елементів, який повертається функцією strel.

Коли IM є логічним або структурним плоским елементом, функція imerode виконує бінарне нарощування; у інших випадках півтонове потоншення. Коли SE є масивом структурних елементів об'єкту, функція imerode виконує багатократне потоншення початкового зображення, використовуючи кожен структурний елемент з послідовності SE.

Функція $IM2 = \text{imerode}(IM, NHOOD)$ виконує операцію потоншення зображення IM, де NHOOD є масивом нулів і одиниць, що описує структурні елементи околиці. Синтаксично ця функція еквівалентна виразу $\text{imerode}(IM, \text{strel}(NHOOD))$. Функція imerode визначає центральний елемент околиці за виразом $\text{floor}((\text{size}(NHOOD)+1)/2)$.

Функції $IM2 = \text{imerode}(IM, SE, \text{PASCORT}, M)$ або $\text{imerode}(IM, NHOOD, \text{PASCORT}, M)$ визначають який масив IM є бінарним зображенням і проводить перетворення його розмірностей M до виду початкового розпакованого зображення. Параметр PASCORT може набувати наступних значень.

У п'ятому блоці створюється маска рухомих, об'єктів. У нашому випадку, це автомобілі. Маска об'єктів на одному з кадрів представлена на

Шостий блок представленої нижче блок-схеми відповідає стеженню за автомобілями і поділом автомобілів по категоріях. Для цього була використана функція `STATS=regionprops(L, properties)`, яка вимірює набір характеристик для кожної області, відміченої в матриці `L`. Позитивні елементи `L` відповідають різним областям. Наприклад, набір елементів `L` рівних 1 відповідають 1-ій області, набір елементів `L` рівних 2 відповідає 2-й області і т.д.



Рис. 2.18. Загальний алгоритм роботи програми визначення інтенсивності транспортних потоків

Результуюча змінна `STATS` має структуру у вигляді масиву довжиною `max(L(:))`. Поля структурного масиву позначають різні параметри для кожного регіону і визначають їх характеристики.

Параметри: 'Area' - скаляр; дійсне число пікселів в позначеній області. (Це значення може трішки відрізнятись від значення, що видається функцією `bwarea`.)

'Centroid' - вектор розмірністю $\text{ndims}(L)$; центр мас області. Відзначимо, що першим елементом характеристики Centroid є горизонтальна координата (або x-координата) центру мас, а другим елементом є вертикальна координата (або y-координата). На Рис. 2.20 представлено маску об'єкта із вказаним синьою зірочкою центром маси.

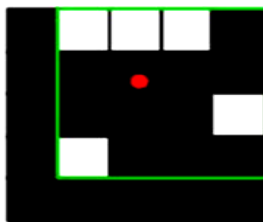


Рис. 2.19. Центр маси об'єкта

Рис. 2.19 предсталає центр мас і обмежуючий прямокутник. Область складається з білих пікселів, зеленим відмічений обмежуючий прямокутник, а червоною точкою відмічений центир мас.



Рис. 2.20. Маска рухомого об'єкту з зазначеним центром мас

Визначити тип автомобілів можна використовуючи площу рухомих об'єктів. Так як легковий автомобіль займає меншу площу як вантажний. Для визначення площі рухомих об'єктів використовується вбудована функція `regionprops(L,Area)`, де `L` це маска.

3. ЛАБОРАТОРНА РОБОТА № 1

Тема: Робота з зображеннями.

Мета: навчитися виконувати елементарні дії з зображеннями, перетворювати матриці зображень.

Завдання:

1. Відкрити відеофайл із записом руху автомобілів.
2. Зберегти з відео файлу серію з 5-ти зображень.
3. Відкрити зображення, з розширенням .jpg;
4. Визначити розмір зображення;
5. Збережіть інформаційні поля зображення в структурні змінні.
6. Перетворіть зображення в чорнобіле.
7. Провести дослідження медіанного фільтру.
8. Змістити два послідовні кадри відносно себе таким чином щоб різниця віднімання цих кадрів була найменшою.
9. Описати всі функції, які використовувалися для вище перелічених завдань.

4. ЛАБОРАТОРНА РОБОТА № 2

Тема: Створення маски рухомих об'єктів

Мета: навчитися виявляти рухомі об'єкти на серії послідовних зображень.

Завдання:

1. Визначити маску рухомих об'єктів на серії послідовних зображень;
2. Визначити центр маси рухомих об'єктів;

5. ЛАБОРАТОРНА РОБОТА № 3

Тема: Трекінг (супроводження) об'єктів

Мета: навчитися встановлювати відповідності між об'єктами або їх частинами на послідовності кадрів, а також визначати їх траєкторію руху і швидкість.

Завдання:

1. Встановити відповідності між об'єктами або їх частинами на послідовності кадрів.
2. Визначати траєкторію руху.
3. Визначити швидкість переміщення об'єктів на серії послідовних зображень у пікселях.

ЖИТЕПАТҮПА

1. Chris Stauffer, W. Eric L. Grimson "Learning Patterns of Activity Using Real-Time Tracking", 2000.
2. Omar Javed, Khurram Shafique, Mubarak Shah "A Hierarchical Approach to Robust Background Subtraction using Color and Gradient Information", 2002.
3. Andrea Prati, Ivana Mikić, Costantino Grana and Mohan M. Trivedi "Shadow Detection Algorithms for Traffic Flow Analysis: a Comparative Study", 2002.
4. Dirk Focken, R. Stiefelhagen "Towards Vision-based 3-D People Tracking in a Smart Room", 2002.
5. Alexandre R.J. François and Gérard G. Medioni "Adaptive Color Background Modeling for Real-Time Segmentation of Video Streams", 2001.
6. L. Wang, W. Hu, and T. Tan. *Pattern Recognition*, 36(3):585–601, 2003.
7. C. R. Wren, A. Azarbayejani, T. J. Darrell and A. P. Pentland. *IEEE Pattern Recognition and Machine Intelligence*, 19(7):780–785, 1997.
8. M. Valera and S. A. Velastin. *IEE Proc.-Vis. Image Signal Process.*, 152(2):192–204, 2005.
9. D. Koller, J. Weber, T. Haung and J. Malik, Moving Object Recognition and Classification based on Recursive Shape Parameter Estimation, In Proceedings of the 12th Israeli Conference on Artificial Intelligence, Computer Vision and Neural Networks, pp. 359-368, Tel-Aviv, Israel, December 1993.
10. Ozawa N., Fujishiro I. A morphological approach to volume synthesis of weathered stones // Volume Graphics workshop '99, Swansea, UK. s 1999. s Vol. 2. s P. 207 s 220.
11. Sikora, T., The MPEG-4 Video Standard Verification Model, *IEEE Transactions, Circuits Systems, Video Technology*, vol. 7, pp. 19-31, Feb. 1997.
12. Nack F. and Lindsay A. T., Everything you Wanted to Know about MPEG-7: Part 2, *IEEE Multimedia*, vol. 6, pp. 64-73, Dec. 1999.

13. Salembier P. and Marques F., Region-based Representations of Image and Video: Segmentation Tools for Multimedia Services, IEEE Transactions, Circuits Systems, Video Technology, vol. 9, pp. 1147-1169, Dec. 1999.
14. Wang D., Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking, IEEE Transactions, Circuits Systems, Video Technology, vol. 8, pp. 539-546, Sept. 1998.
15. Y. Yokoyama, Y. Miyamoto and M. Ohta, Very Low Bit Rate Video Coding Using Arbitrarily Shaped Region-Based Motion Compensation, IEEE Transactions, Circuits System. Video Technology, vol. 5, pp. 500-507, Dec. 1995.
16. L. Wu, J. Benoit-Pineau, P. Delagnes and D. Barba, Spatio-temporal Segmentation of Image Sequences for Object-Oriented Low Bit-Rate Image Coding, Signal Processing: Image Communication., vol. 8, pp. 513-543, 1996.
17. P. Salembier, Morphological Multiscale Segmentation for Image Coding, Signal Processing, vol. 38, pp. 359-386, 1994.
18. N. T. Watsuji, H. Katata and T. Aono, Morphological Segmentation with Motion Based Feature Extraction, In International Workshop on Coding Techniques for Very Low Bit-Rate Video, Tokyo, Nov. 8-10, 1995.
19. W. H. Hong, N. C. Kim and S. M. Lee, Video Segmentation Using Spatial Proximity, Color and Motion Information for Region-Based Coding, In Proceedings SPIE Visual Communications and Image Processing, vol. 2308, pp. 1627-1633, 1994.
20. Choi J. C., Lee, S. W. and Kim, S. D., SpatioTemporal Video Segmentation Using a Joint Similarity Measure, IEEE Transactions, Circuits Systems, Video Technology, vol. 7, pp. 279-289, Apr. 1997.
21. Vibha L., Venugopal K. R. and L. M. Patnaik, A Study of Breast Cancer Using Data Mining Techniques, Technical Report, University Visvesvaraya College of Engineering, Bangalore University, August 2003.
22. Aach T., Kaup A. and Mester R., Statistical Model Based Change Detection in Moving Video, Signal Processing, vol. 31, pp. 165-180, Mar. 1993.

23. Neri A., Colonnese S., Russo G. and Talone P., Automatic Moving Object and Background Separation, *Signal Processing*, vol. 66, pp. 219-232, Apr. 1998.
24. Stauder J., Mech R. and Ostermann J., Detection of Moving Cast Shadows for Object Segmentation, *IEEE Transaction Multimedia*, vol. 1, pp. 65-76, Mar. 1999.
25. D. Koller, J. Weber, T. Haung, J. Malik, G. Ogasawara, B. Rao and S. Russel, Towards Robust Automatic Traffic Scene Analysis in Real-Time, In *Proceedings of the 12th International Conference on Pattern Recognition (ICPR-94)*, pp. 126-131, Jerusalem, Israel, October 9-13, 1994.
26. Puzicha J., Hofmann T. and Buhmann J. M., Histogram Clustering for Unsupervised Image Segmentation, In *IEEE Computer Society Conference Computer Vision and Pattern Recognition*, Fort Collins, CO, pp. 602-608, June 1999.
27. Chen S. C., Shyu M. L. and Zhang C., An Unsupervised Segmentation Framework for Texture Image Queries, In *25th IEEE Computer Society International Computer Software and Applications Conference (COMPSAC)*, Chicago, Illinois, USA, Oct. 2000.
28. Chen S. C., Shyu M. L. and Zhang C., An Intelligent Framework for Spatio-Temporal Vehicle Tracking, *4th International IEEE Conference on Intelligent Transportation Systems*, Oakland, California, USA, Aug. 2001.
29. D. Beymer, P. McLauchlan, B. Coifman and J. Malik, A Real-time Computer Vision System for Measuring Traffic Parameters, In *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*, Puerto Rico, pp. 496-501, June, 1997,
30. O. Masoud and N. P. Papanikolopoulos, Robust Pedestrian Tracking Using a Model-based Approach, In *Proceedings of IEEE Conference on Intelligent Transportation Systems*, pp. 338-343, Nov. 1997.
31. A. J. Lipton, H. Fujiyoshi and R. S. Patil, Moving Target Classification and Tracking from Real-time Video, In *Proceedings of IEEE Workshop Applications of Computer Vision*, pp. 814, 1998.

32. Gupte S., Masoud O., Martin R. F. K. and Papanikolopoulos N. P., Detection and Classification of Vehicles, In IEEE Transactions on Intelligent Transportation Systems, vol. 3, no.1, pp. 37-47, March 2002.
33. Dailey D. J., Cathey F. and Pumrin S., An Algorithm to Estimate Mean Traffic Speed Using Uncalibrated Cameras, In IEEE Transactions on Intelligent Transportation Systems, vol. 1, No. 2, pp. 98-107, June, 2000.
34. Grimson W. E. L., Stauffer C., Romano R. and Lee L., Using Adaptive Tracking to Classify and Monitor Activities in a Site, In IEEE Computer Society Conference on Computer Vision and Pattern Recognition Proceeding, pp. 22-31, 1998.
35. S. Gupte, O. Masoud, R. Martin, and N. Papanikolopoulos, "Detection and classification of vehicles," IEEE Trans. Intelligent Transportation Systems, vol. 3, no. 1, March 2002, pp. 37 – 47.
36. Z. Kim and J. Malik,; "Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking",. Ninth IEEE International Conference on Computer Vision, 13-16 Oct. 2003, pp. 524 – 531.
37. H. Schneiderman, and T. Kanade, "A statistical method for 3D object detection applied to faces and cars," IEEE Conf. Computer Vision and Pattern Recognition, vol.1, 13-15 June 2000, pp. 746 - 751.
38. T. Zhao, and R. Nevatia, "Car detection in low resolution aerial image," Image and Vision Computing, vol. 21, 18 march 2003, pp. 693 – 703.
39. A. Rajagopalan, P. Burlina, and R. Chellappa, "Higher order statistical learning for vehicle detection in images," in Proc. 7th Int. IEEE Conf. Computer Vision, vol. 2, 20-27 Sept. 1999, pp. 1204 -1209.
40. S. Hinz, "Detection and counting of cars in aerial images," in Proc. Int. Conf. Image Processing, vol. 3, 14-17 Sept. 2003, pp. 997-1000.
41. R. Ruskone, L. Guigues, S. Airault, and O. Jamet, "Vehicle detection on aerial images: a structural approach," in Proc. 13th Int. IEEE Conf. Pattern Recognition, vol. 3, 25-29 Aug. 1996, pp. 900 - 904.

42. C. Hoffman, T. Dang, and C. Stiller, "Vehicle detection fusing 2D visual features," in Proc. IEEE Intelligent Vehicles Symposium, 14-17 June 2004, pp. 280 – 285.
43. S. Mantri, and D. Bullock, "A neural network based vehicle detection and tracking system," in Proc. 27th Southeastern Symposium on System Theory, 12-14 March 1995, pp. 279 – 283.
44. D. Ha, J. Lee, and Y. Kim, "Neural-edge-based vehicle detection and traffic parameter extraction," Image and vision computing, vol. 22, 2004, pp. 899-907.
45. Z. Liu, X. Li, and X. Leung, "Fuzzy measures for vehicle detection," The 10th IEEE Int. Conf. Fuzzy Systems, vol. 2, 2-5 Dec. 2001, pp. 848 – 851.

НАВЧАЛЬНЕ ВИДАННЯ

Мельник Михайло Романович

ЛАБОРАТОРНИЙ ПРАКТИКУМ

з курсу “Розпізнавання образів і комп’ютерний зір”

Редактор

Видавництво Національного університету “Львівська політехніка”

Львів – 2025