

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
КАФЕДРА ТЕОРЕТИЧНОЇ КІБЕРНЕТИКИ

Пашко А.О.

МЕТОДИ КЛАСИФІКАЦІЇ ОБ'ЄКТІВ, ЩО РУХАЮТЬСЯ

МЕТОДИЧНІ МАТЕРІАЛИ

до курсу
"МЕТОДИ ОБРОБКИ МУЛЬТИМЕДІЙНОЇ ІНФОРМАЦІЇ"
для студентів

галузь знань	12 «Інформаційні технології»
спеціальність	122 «Комп'ютерні науки та інформаційні технології»
освітній рівень	магістр
освітня програма	«Штучний інтелект»

УДК 519.2:681.3

МЕТОДИ КЛАСИФІКАЦІЇ ОБ'ЄКТІВ, ЩО РУХАЮТЬСЯ /

Пашко А.О. : Електронне видання, -2019.-52 с.

Затверджено Вченою радою

факультету комп'ютерних наук та кібернетики

Протокол №1 від 16 вересня 2019 року

ЗМІСТ	
ВСТУП	4
СЕГМЕНТАЦІЯ ОБ'ЄКТІВ.	7
МЕТОДИ, ЗАСНОВАНІ НА ВИДІЛЕННІ КОНТУРІВ	16
ВИКОРИСТАННЯ КЛАСИФІКАЦІЙНИХ МЕТРИК І ЧАСОВОЇ УЗГОДЖЕНОСТІ	23
МЕТОДИ, ЩО ЗАСНОВАНІ НА ПЕРІОДИЧНОСТІ РУХУ	32
МЕТОДИ, ЩО ЗАСНОВАНІ НА ВИКОРИСТАННІ ДИНАМІЧНИХ ТЕКСТУР	38
ЛІТЕРАТУРА	50

ВСТУП

Відеопотік може містити велику різноманітність рухомих об'єктів. Одні з них представляють інтерес для моніторингу - наприклад, зображення людей, транспортних засобів, тварин і т. п. Інші об'єкти, такі як гілки дерев, тіні, двері, що обертаються, створюють перешкоди при відеоспостереженні. Вся сукупність рухомих об'єктів становить так званий передній план, причому апріорно невідомо число таких об'єктів, якого вони розміру і їх взаємне розташування.

Модуль, який виділяє рухомі об'єкти, називається детектором руху. Для того щоб провести класифікацію об'єктів, що рухаються, потрібно спочатку відокремити кожен об'єкт від інших. Ця операція називається сегментацією об'єктів. Результатом проведення сегментації є або позначений набір пікселів переднього плану, або вказівка вершин мінімально можливого прямокутника (ММП), всередині якого знаходиться виділений об'єкт (рис. 1).

У першому випадку говорять про виділення БЛОБ, який задається набором координат пікселів, що його складають. Поняття БЛОБ було введено в роботах [1,2] для опису компактних структурно значущих областей, в яких кольорові і просторові характеристики пікселів близькі один одному. Для представлення Блоб зручні методи, що засновані на виділенні контурів. Виділення описаного прямокутника (ММП) дає більш грубий опис об'єкта, хоч і вимагає значно меншого часу обробки. Коли об'єкти добре віддільні один від одного, по співвідношенню сторін прямокутника можна попередньо визначити тип об'єкта. Але такий спосіб класифікації вимагає подальшого уточнення. Розглянемо, наприклад, рис. 2. на якому автоматично були детектовані три рухомих об'єкти: людина, автомобіль і група людей. Стосовно сторін прямокутників можна впевнено відрізнити людину від автомобіля, однак за цією ознакою група людей буде класифікована як автомобіль. Аналіз же контурів, зображених на рис. 2В. можливо дозволить уникнути цієї помилки.

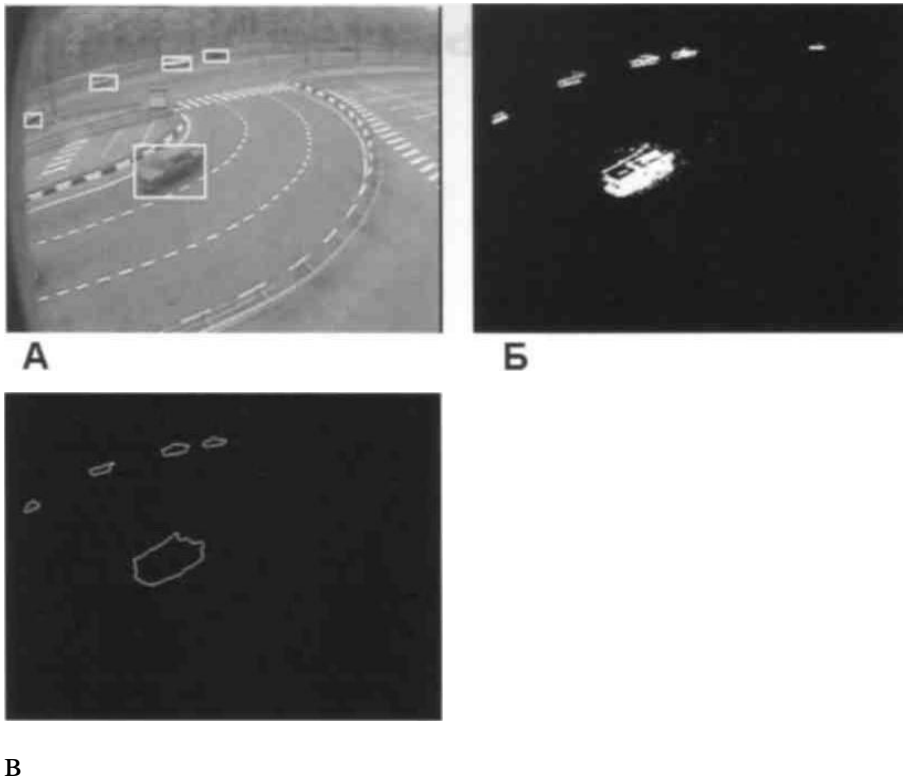


Рис. 1. Виділення рухомих об'єктів: А - за допомогою описаного прямокутника, Б - шляхом виділення «Блоб». В - описом границі Блоб

Практика показує, що після сегментації об'єктів переднього плану, не слід відразу приступати до їх класифікації, а потрібно провести додаткову обробку, мета якої полягає в усуненні дефектів в зображеннях виділених об'єктів. Дійсно, через наявність шумів отримані БЛОБ зазвичай мають всередині порожнини, нерівні границі, і навіть нерідко один об'єкт може бути роздроблений на безліч не пов'язаних між собою фрагментів. Такі вади найпростіше усунути шляхом виконання відповідних морфологічних операцій, наприклад, дилатації і подальшої ерозії. При правильному виборі для цих операцій структуруючого елемента цілісність об'єктів буде відновлена, а контури об'єктів стануть більш плавними. Оскільки аналіз відеоряду, як правило, проводиться в масштабі реального часу, алгоритми класифікації повинні бути відносно швидкими, ефективними і інваріантними до постійно змінюваних умов освітлення. Аналіз великої кількості методів, які відповідають цим умовам, показав, що всі вони можуть бути розділені на

три великі групи: геометричні методи, методи, що засновані на динамічних характеристиках руху, і методи, що засновані на динамічних структурах. Перші засновані на виділенні ряду ознак, що характеризують геометричну форму об'єкта, накопичення такої інформації за певний період часу, створення бази даних чисельних шаблонів (під час періоду навчання) і порівняння об'єкта, що цікавить, з шаблонами. До першої групи належать методи, що засновані на сегментації об'єктів і різні контурні методи.

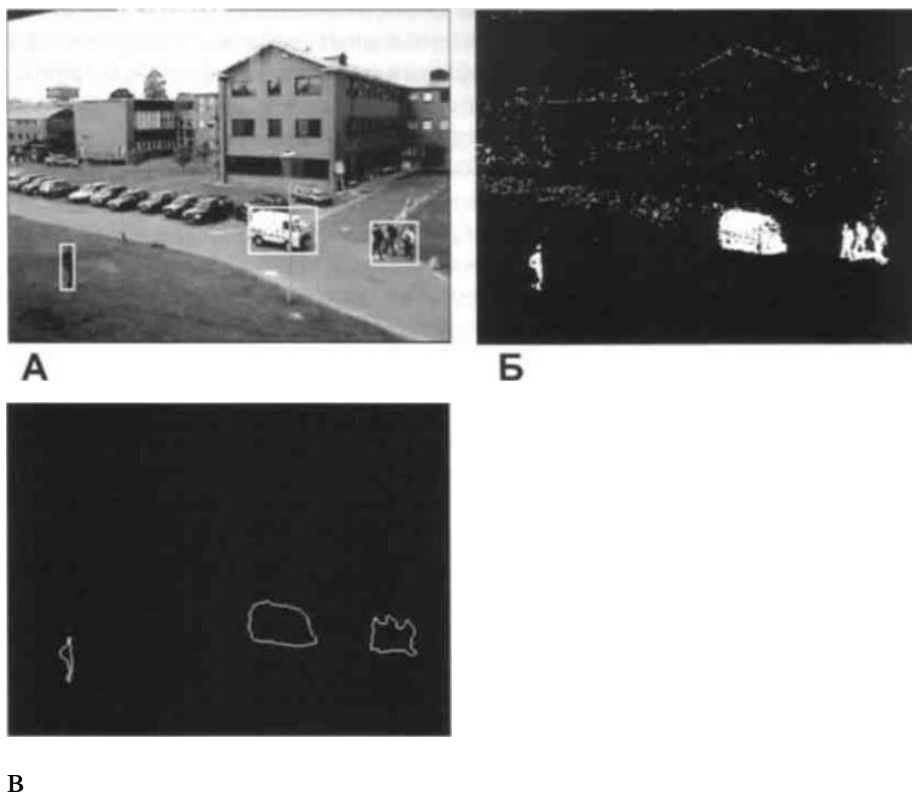


Рис. 2. Виділення рухомих об'єктів: А - за допомогою описаного прямокутника, Б - шляхом виділення Блоб, В - описом кордону Блоб

Друга група методів використовує періодичність рухових процесів, що присутні практично у всіх об'єктах (наприклад, обертання колеса автомобіля). Один з можливих алгоритмів - стійка до шумів технологія визначення характеристик оптичного потоку (поля швидкостей). Однак її головним недоліком є високі вимоги до ресурсів комп'ютера. І, нарешті, третя група методів базується на використанні динамічних текстур (ДТ) - тривимірних варіантів звичайних двовимірних текстур. В якості додаткового

виміру в ДТ виступає час. Застосування ДТ для класифікації об'єктів є одним з найбільш перспективних і багатообіцяючих в даний час. Далі розглядаються всі етапи обробки рухомих об'єктів, а також різноманітні методи отримання характерних ознак, які в подальшому можуть бути використані для побудови системи, що розпізнає.

СЕГМЕНТАЦІЯ ОБ'ЄКТІВ.

Сегментацією називається процес розбиття зображення на складові його об'єкти. Це завдання ретельно вивчається в більшості книг, присвячених цифровій обробці зображень. Що стосується завданням обробки відеоряду виникає ряд особливостей, як полегшують її рішення, так і ускладнюють. Оскільки процесу сегментування піддаються тільки рухомі об'єкти, сумарна площа яких зазвичай істотно менше площі, що зайнята всім зображенням, то обсяг оброблюваної інформації менше, ніж при сегментуванні статичних зображень.

До того ж рухомі об'єкти часто бувають ізольованими один від одного. Ці фактори полегшують рішення поставленого завдання. До обставин, що ускладнюють, можна віднести проблему збору спостережуваних об'єктів з частин, на які вони були поділені в результаті роботи детектора руху. Така проблема може виникнути, наприклад, в тому випадку, якщо частина рухомого об'єкту має колір, що співпадає з кольором фону, в результаті чого в цьому місці можуть з'явитися порожнечі химерної форми. А головною вимогою до алгоритму зазвичай є можливість його використання в реальному часі, що накладає дуже жорсткі обмеження [3,4].

Дуже ефективним з точки зору обчислювальних ресурсів є використання для сегментації методів кластерного аналізу. Суть кластеризації полягає в тому, що всі вихідні об'єкти (в даному випадку пікселі) розбиваються на кілька непересічних груп таким чином, щоб об'єкти, що потрапили в одну групу, мали подібні характеристики. В той час як у об'єктів з різних груп ці характеристики повинні значно відрізнятися. Отримані групи називаються

кластерами. Вихідними даними для кластеризації в разі напівтонових зображень є тривимірні вектори $(x, y, I(x, y))$, де x, y - координати пікселя в довільній системі координат, $I(x, y)$ - його інтенсивність. У разі використання кольорових зображень вектори стають п'ятивимірними. В результаті такого кодування в один кластер повинні потрапити групи пікселів, що розташовані близько один до одного і при цьому мають близьку інтенсивність (в разі напівтонових зображень) або схожий колір. Алгоритми побудови кластерів можна розбити на дві групи [5].

Алгоритми першої групи застосовуються в тому випадку, якщо нам апріорно відомо число кластерів K . Типовим представниками цієї групи є об'єднуючий (агломеративний), розділюючий алгоритми, а також алгоритми K -середніх. Для алгоритмів другої групи зазвичай задаються обмеження на розмір створюваних кластерів, а саме число кластерів автоматично визначається в процесі роботи алгоритму. Найбільш ефективним алгоритмом цього типу є метод адаптивного резонансу Гроссберга (ART, Adaptive Resonance Theory).

Агломеративні методи. Основна ідея цих алгоритмів полягає в послідовному об'єднанні окремих елементів і вже побудованих кластерів в більші кластери. У початковий момент вважається, що кожен елемент навчальної множини утворює свій кластер, тобто, в початковий момент ми маємо N кластерів, кожен з яких містить рівно один елемент. Далі вибираються два кластери з найменшою відстанню між ними. Ці кластери об'єднуються в новий, після чого число кластерів стає $N - 1$. Описаний процес повторюється до тих пір, поки число кластерів не стане рівним заданому числу K .

Розділюючі методи. Ці методи засновані на принципах, які діаметрально протилежні принципам, що лежать в основі агломеративних методів, а саме, на розподілі великих кластерів на більш дрібні. У початковий момент передбачається, що всі об'єкти належать одному кластеру. Після цього в ньому знаходяться два об'єкти з найбільшою відстанню між ними. Вони

вважаються центрами двох нових кластерів. Далі знаходиться найбільший кластер, який ділиться на дві частини таким же способом. Процес поділу зупиняється при досягненні заданого числа кластерів.

Алгоритм К-середніх (k-means). У вихідній множині вибираються довільним чином K об'єктів, які вважаються центрами шуканих кластерів, де K - задане число кластерів. Потім проводиться розподіл всіх елементів, що залишилися в множині цим кластерам за принципом близькості до відповідного центру. Далі проводиться уточнення положення центрів, які обчислюються як середнє по сукупності всіх об'єктів, що потрапили в кожен кластер. Цей процес повторюється до тих пір, поки зміна центрів кластерів у порівнянні з обчисленням на попередній ітерації не стане досить малою.

Кластеризація методом ART при невідомому числі кластерів. В цьому випадку необхідно ввести параметр ε - обмеження на взаємну віддаленість сусідніх об'єктів, що належать одному кластеру. З вихідної множини довільним чином вибирається один об'єкт, який призначається центром першого кластера. Далі для кожного елемента множини обчислюється відстань до центрів вже побудованих кластерів. Якщо для будь-якого кластера ця відстань менше значення ε , то елемент приписується цьому кластеру і проводиться уточнення положення центроїда з урахуванням нового члена. Якщо ж новий елемент не потрапляє ні в один кластер, він стає центром нового кластера.

Відзначимо, що алгоритми К-середніх і ART є найбільш ефективними в обчислювальному плані, тому вони можуть бути використані в задачах реального часу. При використанні алгоритму К-середніх слід задати число кластерів, яке, взагалі кажучи, в задачах відеоспостереження зазвичай апріорно невідоме. В цьому випадку можна поступити наступним чином: призначають цьому числу значення, що перевищує число об'єктів, що присутні на сцені, що фіксуються конкретною камерою. Після побудови кластерів їх слід оточити описаними прямокутниками і об'єднати в один кластер об'єкти, для яких прямокутники перетинаються.

На рис. 3. наведені результати сегментації рухомих об'єктів за допомогою алгоритму К-середніх. Максимальне число кластерів було вибрано рівним дев'яти. На рис. 3А зображені знайдені кластери, а на рис. 3Б – кластери, що отримані після об'єднання об'єктів, для яких описані прямокутники перетинаються. Рис. 3Б практично не відрізняється від рис. 1А, який отриманий при використанні методу сегментації, що заснований на виділенні контуру.

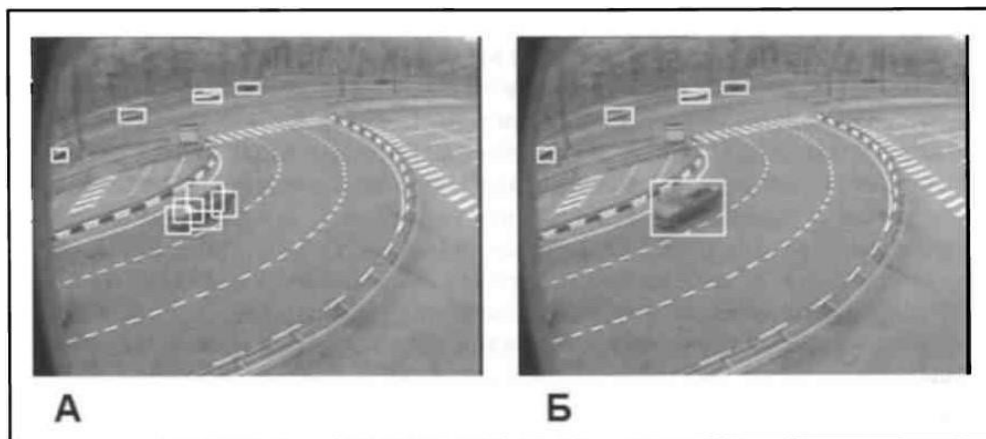


Рис. 3. Виділення рухомих об'єктів за допомогою алгоритму К-середніх: А - після знаходження 9 кластерів, Б - після об'єднання об'єктів з прямокутниками, що перетинаються

ПОСТОБРОБКА ЗОБРАЖЕНЬ ОБ'ЄКТІВ

Після сегментації зображення часто має ряд недоліків у вигляді дрібних областей, ліній і окремих пікселів, не віднесені до жодної області. Існують різні фактори, які призводять до таких недоліків в сегментованому зображенні:

- інтенсивність пікселя, що знаходиться на границі двох забарвлених в різний колір об'єктів, може бути віднесена до одного кольору в початковому кадрі і до іншого - в наступному кадрі;

- коли джерело світла, наприклад, сонце, рухається, то при цьому деякі частини в фоновому зображенні відбивають світло. Це призводить до того що алгоритми виділення фону невірно сегментують і відбивають світло області в передній план;

- деякі частини об'єктів можуть бути пофарбовані в колір фону. В результаті вони будуть виключені з переднього плану;

- тіні, що падають на об'єкти, більшістю алгоритмів сегментуються в передній план. Крім того раптова зміна освітленості сцени також призводить до неправильної сегментації зображення.

Залежно від характеру недоліка, на сегментованому зображенні слід застосовувати процедури злиття областей і операції морфологічної фільтрації зображення.

Морфологічна фільтрація. До бінарного зображення застосовуються послідовно такі операції математичної морфології, як розширення (дилатація) і стиснення (ерозія). Нехай підмножина S - сукупність точок зображення, що належать об'єкту, що цікавить, B - структуруючий елемент. Операція ерозії визначається в математичній морфології наступним чином [6]:

$$\varepsilon_B(S) = S - B = \{B_h \subseteq S | h \in E\} \quad (1)$$

де $\varepsilon_B(S)$ означає ерозію підмножини S із структуруючим елементом B ; E - дискретна площина; h – вектор, що визначає кожну позицію B на E ; B_h - перенесення B вздовж вектора h :

$$B_h = \{b + h | b \in B\}. \quad (2)$$

Під час процедури ерозії елементи не видаляються із зображення, а лише позначаються для видалення після закінчення всієї процедури. Приклад застосування операції ерозії наведено на рис. 4.

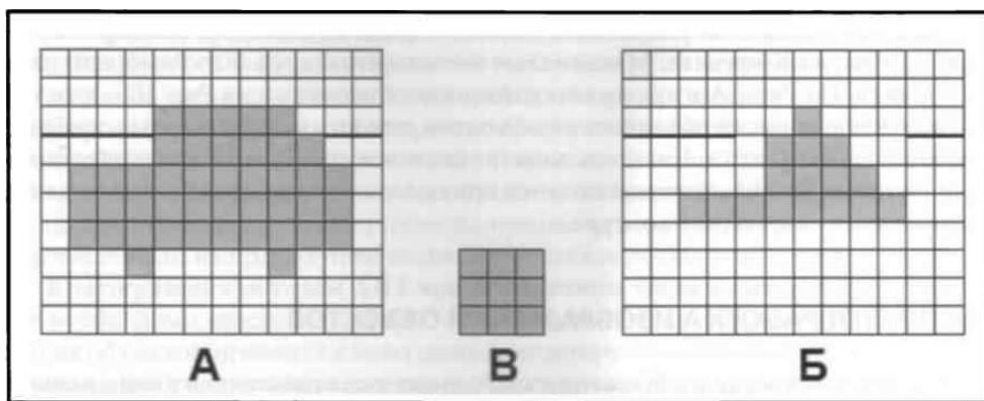


Рис. 4. Приклад застосування операції ерозії: А - початкове зображення, Б - результат застосування операції ерозії до зображення А, В - структуруючий елемент

Розширення (дилатація) визначається як:

$$\delta_B(S) = S \oplus B = \{s + b | s \in S, b \in B\}, \quad (3)$$

де $\delta_B(S)$ означає розширення підмножини S із структуруючим елементом B . Приклад використання операції дилатації приведено на рис. 5.

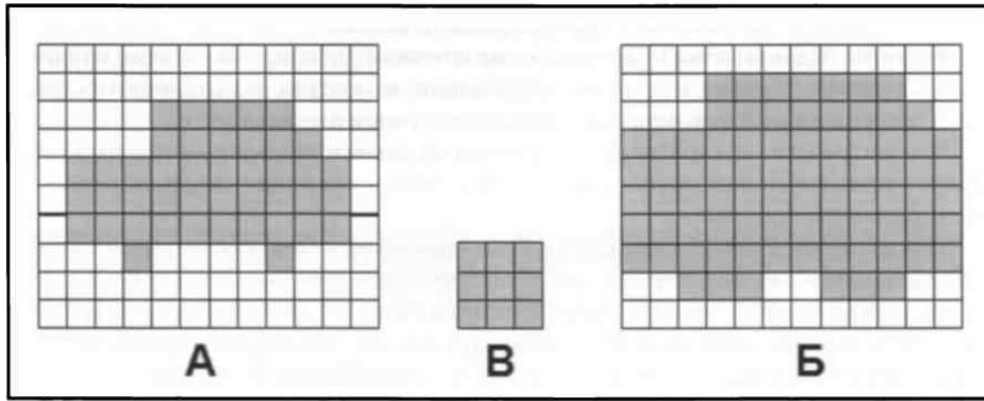


Рис. 5. Приклад застосування дилатації: А – початкове зображення, Б - результат застосування операції дилатації до зображення А, В - структуруючий елемент.

Послідовне застосування ерозії і дилатації веде до згладжування бінарних зображень [7].

Результат обробки істотно залежить від двох чинників: зміни структуруючого елемента і його розміру [8]. Як відомо, вибір конфігурації є скоріше емпіричним і інтуїтивним процесом. Структуруючий елемент для ерозії може являти собою, наприклад, сукупність далеких сусідів опорного пікселя, а для розширення - повну множину сусідів опорного пікселя. Що стосується розміру структуруючого елемента, то очевидно, що операції ерозії і дилатації повинні використовувати один і той же розмір, в іншому випадку товщина ліній об'єкта зміниться після закінчення процесу обробки.

Однак коректний вибір розміру структуруючого елемента для конкретного зображення або його окремих частин є завданням оптимізації, рішення якого залежить від ряду факторів. По-перше, бажано звільнити зображення від дефектів настільки, наскільки методи математичної морфології дозволяють це зробити в принципі, по-друге, одним з основних умов є збереження топології об'єкта, що цікавить, по-третє, час обробки повинно бути досить малим.

Злиття. Маленька область, кількість пікселів якої менше заданого порогу, приєднується до найближчої до неї по яскравості великої області. Одиночний піксель додається до найближчої до нього по яскравості області,

або аналізується приналежність пікселів його околиці до сусідніх областей, і приймається рішення про його додавання до тієї області, до якої належить більшість його сусідів. Приклад злиття областей зображений на рис. 6.

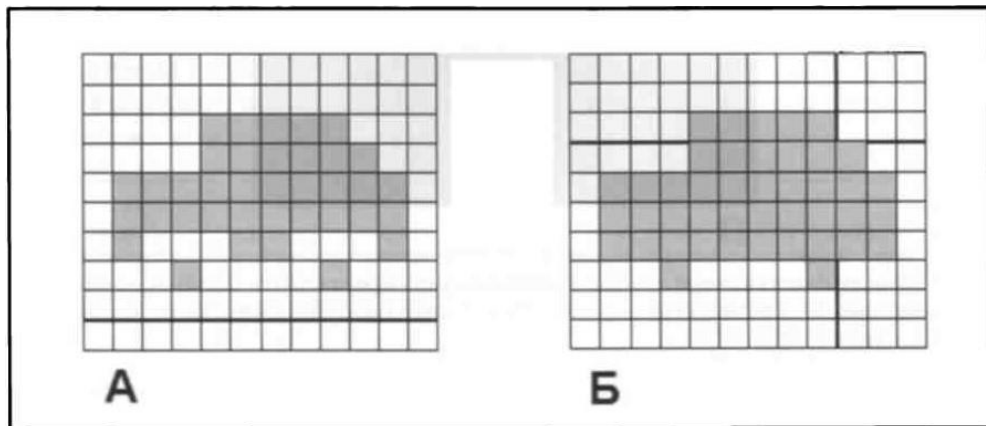


Рис. 6. Приклад злиття областей: А – початкове зображення з відокремленими «колесами» автомобіля, Б - зображення, що вийшло в результаті злиття областей.

Цей вид постобробки рекомендований для алгоритмів нарощування областей та алгоритмів із застосуванням кластерного аналізу, так як часто в результаті їх роботи виходять дрібні області або пікселі, що не віднесені до жодної області. Найпростіше злиття областей виконати шляхом застосування морфологічної операції замикання, яка визначається як

$$S \bullet B = (S \oplus B) - B.$$

Цей запис потрібно розуміти в такий спосіб. До множини S застосовується операція дилатації зі структуруючим елементом B , а до отриманого результату застосовується операція ерозії з тим же структуруючим елементом B .

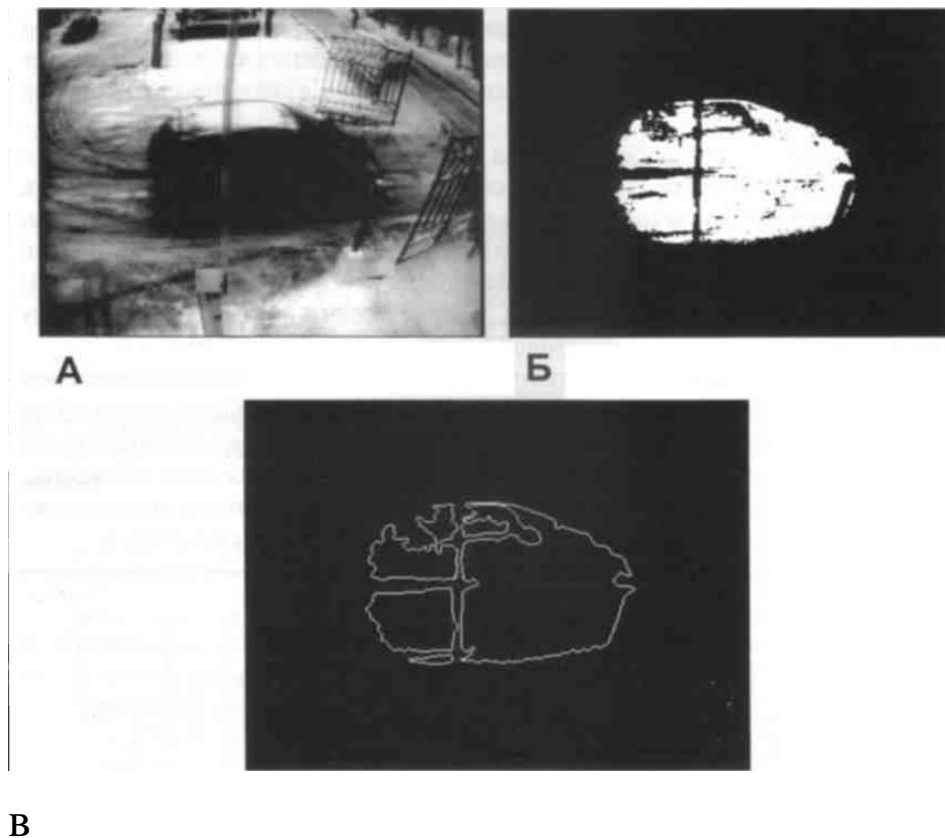


Рис. 7. Між рухомим автомобілем і камерою знаходиться стовп. А – початкове зображення, Б - відповідний блоб, який розділився на частини, В - результат оконтуровання Блоб

При відеоспостереженні часто виникає ситуація, коли між об'єктом, що рухається і камерою знаходиться яка-небудь перешкода, наприклад, дерево або стовп. В результаті сегментування цей об'єкт розбивається на дві частини, що може привести до помилки при класифікації. Ця ситуація зображена на рис. 7. Видно, що коли автомобіль проїжджає повз стовп, що розташований ближче до камери, то відповідний автомобілю блоб розтинають стовпом на дві частини. Оскільки область, яку загороджує стовп, має незначні в порівнянні з автомобілем розміри, то застосування операції злиття дозволяє відновити зображення БЛОБ.

МЕТОДИ, ЗАСНОВАНІ НА ВИДІЛЕННІ КОНТУРІВ

Проблемі виділення контурів присвячена велика кількість літератури. Завдання обробки відеозображення в реальному часі накладає на обраний алгоритм суттєві обмеження по продуктивності, тому з усіх алгоритмів слід в першу чергу вибирати найбільш швидкі, можна, навіть на шкоду якості. На практиці добре себе зарекомендував метод, що полягає в послідовному проходженні бінарного зображення зверху вниз і зліва направо, знаходженні пов'язаних областей (далі компонент) і присвоєння їм міток [9].

При цьому всі операції можна умовно розділити на чотири етапи.

1) Знаходиться перша точка зовнішнього контуру, яка на рис. 8А позначена буквою А, після чого весь зовнішній контур проходиться цілком, поки знову не зустрінеється точка А. Їй і всім точкам зовнішнього контуру присвоюється одна і та ж мітка.

2) Для всіх точок зовнішнього контуру, що перебираються зверху, виконується наступна операція. Починаючи з цієї точки, позначеної на рис. 8Б як А', здійснюється сканування відповідного рядка пікселів, щоб визначити всі наступні чорні пікселі (якщо вони існують) шляхом присвоєння їм тієї ж мітки, що і у А'.

3) Коли в перший раз зустрічається точка внутрішнього контуру В (рис. 8Б,В), їй присвоюється та ж мітка, що і у відповідної точки того ж ряду А' зовнішнього контуру. Далі обходиться весь внутрішній контур, і всім його точкам присвоюється та ж мітка, що і у точки В (рис. 8В).

4) Коли зустрічається точка В' внутрішнього контуру, що вже має мітку (рис. 8Г), то продовжуємо йти вздовж цього ряду, щоб визначити всі наступні чорні пікселі (якщо такі існують) і присвоїти їм ту ж мітку, що і у В'.

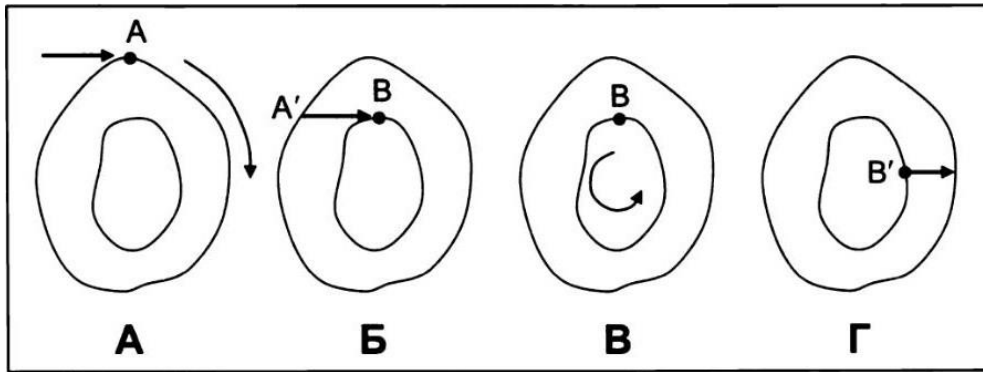


Рис. 8. Послідовність побудови контуру

Для простоти припустимо, що перший рядок зображення складається з білих пікселів (якщо це не так, то до зображення можна додати ще один рядок з білих пікселів). Створимо робочий масив тієї ж розмірності, що і вихідне зображення, і будемо в ньому зберігати мітки кожного пікселя. Позначимо через C індекс мітки для компонент: нехай в початковий момент він має значення 1. Пікселі, яким будуть присвоєні однакові мітки, належатимуть одному знайденому об'єкту, а номер мітки буде відповідати номеру цього об'єкта. Позначимо буквою P поточний піксель, що аналізується.

1 етап. Виділення зовнішнього контуру. Якщо у пікселя P немає мітки і піксель над ним білий, як це показано на рис. 9, це означає, що P лежить на зовнішньому контурі. Пікселю P присвоюється мітка C , після чого здійснюється перехід до процедури виділення контуру *Contour Tracing*, що описується нижче. В результаті виконання якої всім точкам цього контуру присвоюється мітка C . Після виконання процедури *Contour Tracing* значення індексу C збільшується на 1.

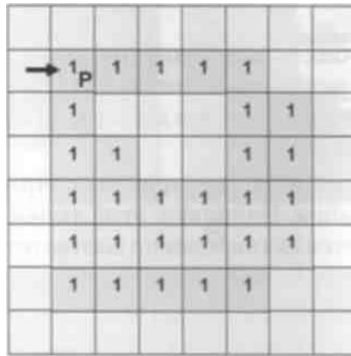


Рис. 9. P - початкова точка зовнішнього контуру, 1 - непомічені чорні пікселі.

2 етап. Якщо піксель під P є відкритим білим пікселем, то P лежить на новому внутрішньому контурі. При цьому існує дві альтернативи:

- P вже має мітку (рис. 10А). В цьому випадку P також є точкою і зовнішнього контуру.
- У P немає мітки (Рис. 10Б). У цьому випадку попередній піксель N перед P в цьому рядку повинен мати мітку. Тоді P отримує ту ж мітку що і N.

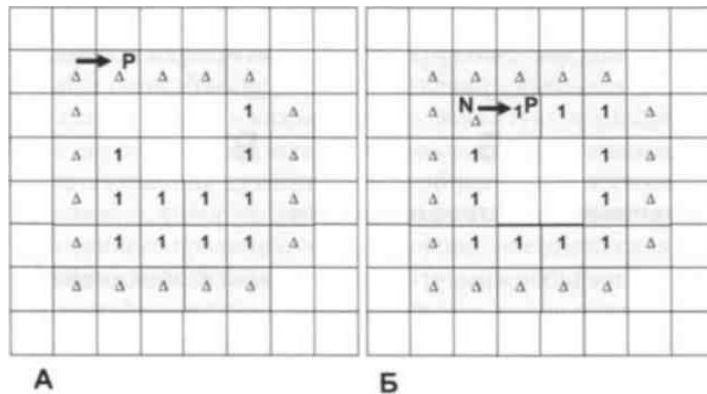


Рис. 10. А. P - початкова точка внутрішнього контуру; P також належить зовнішньому контуру. Б. P - початкова точка внутрішнього контуру: P не належить зовнішньому контуру. 1 - непомічені чорні пікселі, Δ - помічені.

Далі в будь-якому випадку переходимо до *Contour Tracing*, щоб визначити весь внутрішній контур, якому належить P, і присвоїти йому ті ж мітки.

3 етап. Якщо Р не є точкою, що розглядалась на попередніх кроках (тобто, вона не належить ніякому контуру), то в цьому випадку точка зліва від Р повинна мати мітку (Рис. 11). Пікселю Р присвоюється та ж сама мітка.

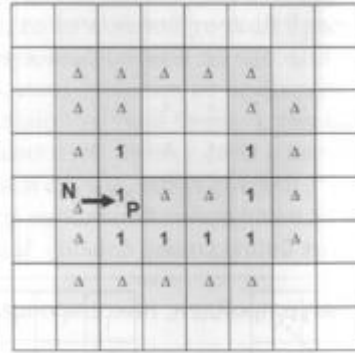


Рис. 11. Точка Р не належить ніякому контуру

Для того щоб уникнути повторного обходу контурів, білі пікселі, навколо помічених точок, позначаються як закриті. Ця ситуація зображена на рис. 12, на якому білі пікселі, що навколо точки Q, позначені як закриті. Тому, коли вздовж рядка ми дійдемо до точки Q, то білий піксель внизу вже не буде відкритим, тому не відбудеться перехід до процедури *Contour Tracing*. У той же час піксель, що розташований під пікселем, що зустрівся вперше, у внутрішньому контурі Р на Рис. 12. все ще є відкритим, так як внутрішній контур, що містить цей піксель, ще не був пройдений.



Рис. 12. *Закриті* білі пікселі позначені ‘-’, *відкриті* – порожні.

Закриваючи навколишні білі пікселі, ми тим самим забезпечуємо проходження кожного внутрішнього контуру тільки один раз. Як впливає з Рис. 13, коли внутрішній контур пройдено, піксель під R вже більше не є відкритим, і тому нам не треба буде проходити внутрішній контур ще раз, коли вздовж рядка ми дійдемо до R (необхідно проходити внутрішній контур тільки в тих точках, коли білі пікселі під ними є відкритими).



Рис. 13. Коли внутрішній контур пройдено, піксель під R вже більше не є відкритим.

Процедра Contour Tracing.

Метою даної процедури є визначення всіх точок внутрішніх і зовнішніх контурів, починаючи з якоїсь точки S. Якщо S визначається як ізольована точка (тобто, серед її восьми сусідів немає чорних точок), то здійснюється вихід з процедури. В іншому випадку знаходиться точка контуру T, що є наступною за S, далі знаходиться точка, що наступна за T і т. д. Процедура виділення контуру триває до тих пір, поки не будуть виконані наступні дві умови одночасно:

1. процедура дійшла до вихідної точки S.
2. наступною за нею точкою контуру буде точка T.

На рис. 14. наведено приклад контуру, який може бути знайдений за допомогою описаної процедури навіть у такому досить непростому випадку, коли він складається з однієї лінії, причому пошук починається з середньої точки контуру.

Якщо не вимагати виконання умови 2, то був би знайдений контур STUTS, що становить лише частину від вірного контуру.

Кожна наступна точка контуру шукається серед восьми сусідніх точок даного пікселя. Нехай, наприклад, нам треба знайти точку наступну за точкою Р.

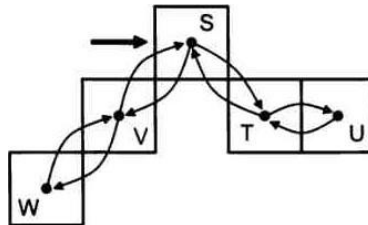


Рис. 14. У даному випадку контур буде таким: *STUTSVWVS*.

Кожній з цих восьми точок присвоюється індекс, як показано на Рис. 15А. Пошук проводиться за годинниковою стрілкою, причому початок обходу визначається наступним чином.

Якщо Р лежить на новому зовнішньому контурі, то початковою точкою буде точка з індексом 7, так як точка над Р є білим пікселем, а наступною точкою за годинниковою стрілкою буде саме точка з номером 7. Якщо ж Р лежить на новому внутрішньому контурі, то початковою точкою буде точка 3, так як точка під Р - білий піксель, а наступною точкою за годинниковою стрілкою буде точка з індексом 3. З іншого боку, якщо існує попередня точка контуру, і це, наприклад, точка 3, то початковою точкою буде 5, так як точка 4 вже повинна була бути пройдена (спочатку йдемо зверху вниз і зліва направо). Ця ситуація показана на Рис. 15Б. У загальному випадку, коли з точки Р не почався ні зовнішній, ні внутрішній контур, номер початкової точки для здійснення обходу визначається як $(d+2) \bmod 8$, де d - номер попередньої точки контуру.

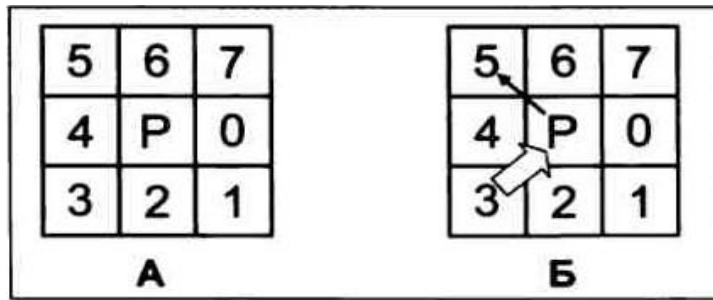


Рис. 15. А - нумерація пікселів навколо поточної точки Р, Б - вибір початкової точки для обходу сусідів за годинниковою стрілкою.

Після того як початкова точка обходу сусідів визначена, за годинниковою стрілкою знаходиться перший чорний піксель. Це і буде точка, наступна за Р. Якщо таку точку знайти не вдається, то Р є ізольованою точкою.

Закриття навколишніх білих пікселів проводиться під час пошуку наступної точки. Як впливає з Рис. 16, якщо А - поточна точка, а С - наступна точка контуру, і при перевірці сусідів точки А до знаходження чорної точки С був знайдений білий піксель В, то він закривається.

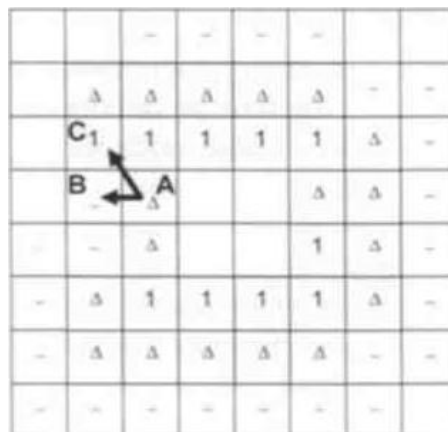


Рис. 16. При переході від однієї точки контуру А до іншої точки контуру С білий піксель В, що зустрівся при скануванні, закривається.

ВИКОРИСТАННЯ КЛАСИФІКАЦІЙНИХ МЕТРИК І ЧАСОВОЇ УЗГОДЖЕНОСТІ

Методи, що описуються, засновані на двох ключових поняттях: класифікаційної метрики (що дозволяє визначити міру близькості даного об'єкту до одного з шаблонів, що зберігаються в створеній базі даних) і часової узгодженості (тобто, класифікуються тільки ті об'єкти, які присутні в відеопотоці протягом ряду послідовних кадрів; а інакше вони вважаються фоновим шумом). Особливі труднощі при класифікації створюють такі обставини: в поточному кадрі справжня природа об'єкта може бути спотворена внаслідок накладання декількох об'єктів. Наприклад, група людей, що йдуть поруч, в окремих кадрах може виглядати як їде автомобіль. Для вирішення цієї проблеми можна використовувати підхід, заснований на породженні і перевірці ряду гіпотез [10-14].

На першому кроці для t -ого кадру за допомогою методу часової різниці виділяється $N(t)$ потенційних об'єктів $F_t(i)$. Ці об'єкти класифікуються за допомогою вибраного метричного оператора $d(x)$, а результат відзначається як гіпотеза $\chi(i)$

$$\chi(i) = \{d(F_t(i))\}, \quad i = 1, \dots, N(t) \quad (4)$$

Далі в дію вступає принцип часової узгодженості: кожен з цих потенційних об'єктів повинен бути присутнім в наступних кадрах для продовження класифікації. Тому в новому кадрі шукається відповідність всіх передньопланових областей попереднього кадру $F_t(i)$ передньоплановим областям поточного $F_{t+1}(i)$ за допомогою метрики $d(x)$. Всі потенційні передньопланові об'єкти з попереднього кадру, яким не відповідає жоден об'єкт поточного кадру, викидаються з подальшого розгляду.

Передньопланові об'єкти поточного кадру, що не відповідають ніякому об'єкту з попереднього кадру, вважаються новими потенційними об'єктами.

Таким чином, в кожному кадрі відбувається оновлення класифікаційної гіпотези:

$$\chi(i) = \chi(i) \cup \{d(F_t(i))\}. \quad (5)$$

Тим самим можуть бути накопичені статистичні дані про об'єкт за певний проміжок часу, що дозволить прийняти правильне класифікаційне рішення. Більш того, такі сторонні рухи, як хитання гілок дерева, будуть виключені з розгляду.

Вибір метричного оператора ґрунтується на тій обставині, що люди, як правило, мають менші розміри і більш складні контури, ніж транспортні засоби. Для обліку цієї закономірності досить ефективним є наступний метод. Введемо систему координат, в якій по одній осі будемо відкладати розмір об'єкта A (загальна кількість пікселів), а по іншій D - величину d , що визначається за такою формулою:

$$D = \frac{p^2}{A} \quad (6)$$

де p - це периметр об'єкта. Після цього на основі даних, що отримані під час навчання системи, на координатній площині можна виділити області, що відповідають людям і транспортним засобам. Подальший поділ між цими двома класами проводиться шляхом розбиття масиву даних або лінійним чином, або за допомогою міри Махаланобіса. На Рис. 17 наведені зображення автомобілів і людей, під якими стоїть значення параметра D , округлене до цілого значення.

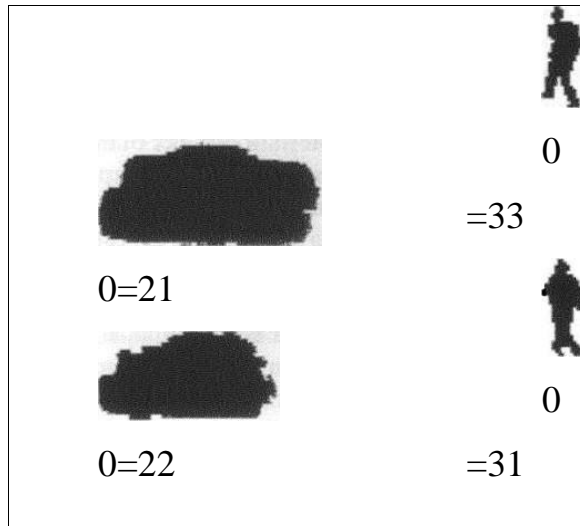


Рис. 17. Значення параметра D для деяких зображень автомобілів і людей

Можна побачити, що для зображень людей параметр D приймає значення, що приблизно в півтора рази більші, ніж для зображень автомобілів. В той час як загальне число пікселів A для автомобілів, як правило, на порядок більше, ніж для присутніх на тій же сцені людей. В результаті цього області, що відповідають зображенням автомобілів і людей, будуть досить далеко рознесені на площині (A, D) , як це показано на Рис. 18.

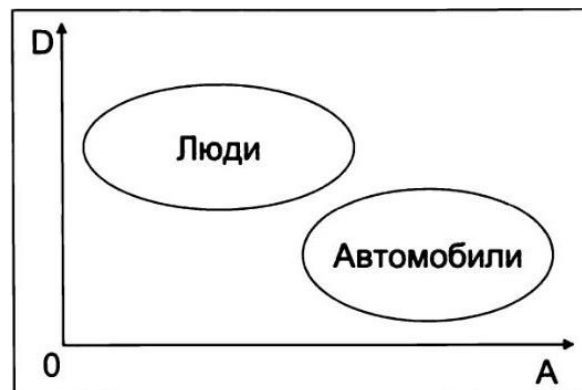


Рис. 18. Взаємне розташування областей, що відповідають зображенням людей і автомобілів на площині (A, D)

Щоб уникнути невірної класифікації об'єктів під час їх перекриття або в тих випадках, коли об'єкт на зображенні з'явився тільки частково (при цьому автомобіль може виглядати як людина), можна використовувати метод максимальної правдоподібності. Для кожного передньопланового об'єкта

слід побудувати класифікаційну гістограму, у якій число осередків збігається з числом різних типів об'єктів. Наприклад, якщо вважати, що в відеопотоці можуть бути присутніми автомобілі, групи людей і окремі люди, то розмір такої гістограми буде дорівнювати трьом, а кожен її осередок буде відображати кількість разів, коли даний об'єкт був віднесений до конкретного класу. Вводиться час класифікації t_{cl} , після якого об'єкт вважається віднесеним до класу, що має найбільше значення в гістограмі. При $t > t_{cl}$ класифікація об'єкта може бути змінена.

Аналогічний підхід полягає в створенні на першому етапі бази даних шаблонів контурів різних об'єктів, яка в подальшому використовується для аналізу відеопотоку. У створену базу даних можуть входити контури людей в різних позах, автомобілі, зняті в різних ракурсах і т. п. При цьому зберігатися будуть не самі зображення, а їх характерні відстані. Нехай $P = \{p_1, p_2, \dots, p_n\}$ - контур об'єкта F , що складається з n точок, які впорядковані за годинниковою стрілкою, починаючи від верхньої центральної точки. Позначимо через $C_m = (x_m, y_m)$ центр мас об'єкта F :

$$x_m = \frac{\sum_{i=1}^n x_i}{n}, \quad y_m = \frac{\sum_{i=1}^n y_i}{n}. \quad (7)$$

Тоді компоненти вектора характерних відстаней об'єкта $\vec{D} = \{D_1, D_2, \dots, D_n\}^T$ визначаються як відстань від центру мас до відповідної точки контуру:

$$D_i = d(C_m, p_i), \quad i = 1, \dots, n \quad (8)$$

де $d(a, b)$ являє собою евклідову відстань між точками $a(x_a, y_a)$ і $b(x_b, y_b)$.

Процес знаходження характерних відстаней об'єкта пояснюється на Рис. 19.

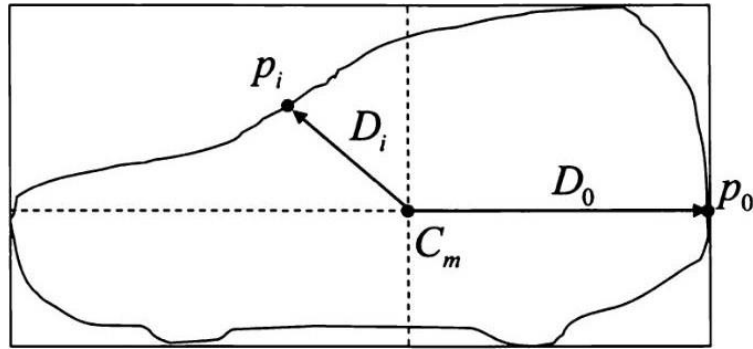


Рис. 19. Обчислення характерних відстаней об'єкта. C_m - центр мас об'єкта, D_i - відстань від центру мас до точки контуру p_i .

Так як різні об'єкти мають відмінні розміри, то, природно, і розміри їх контурів теж не збігаються. Більш того, для одного і того ж об'єкта розміри контуру можуть змінюватися з часом. Тому для того щоб мати можливість порівнювати об'єкти різного розміру, необхідно мати метрику не залежну від масштабу. З цією метою введемо постійну цілу величину Q , що дозволяє фіксувати розмір вектора характерних відстаней об'єкта, в такий спосіб:

$$\hat{D}_i = D \begin{bmatrix} n \\ i \\ Q \end{bmatrix}, \quad i = 1, \dots, Q \quad (9)$$

Тепер можна нормувати отримані величини:

$$\bar{D}_i = \frac{\hat{D}_i}{\sum_{i=1}^n \hat{D}_i} \quad (10)$$

Міра відповідності даного об'єкту F і шаблону T з бази даних визначається як

$$d_{FT} = \sum_{i=1}^Q \left| \bar{D}_i^F - \bar{D}_i^T \right| \quad (11)$$

Щоб визначити, до якого класу належить об'єкт, необхідно знайти дану відстань для кожного шаблону з бази даних. Той шаблон, для якого ця величина буде мінімальною, вважається таким, що має найкращу

відповідність даному об'єкту. Для дотримання принципу часової узгодженості будується класифікаційна діаграма [10].

В якості характерних ознак об'єкта можна розглядати інваріантні моменти [11]:

$$m_{p,q} = \sum_{i=1}^M \sum_{j=1}^N x_i^p y_j^q F(x_i, y_j), \quad p, q = 0, 1, 2, \dots \quad (12)$$

де $F(a, b)$ є передньоплановий об'єкт, а саме, блоб зображення. Однак таке визначення моменту не є інваріантним. Інваріантним до переміщення буде центральний момент $(p+q)$ -порядку:

$$V_{p,q} = \sum_{i=1}^M \sum_{j=1}^N (x_i - x_m)^p (y_j - y_m)^q F(x_i, y_j), \quad p, q = 0, 1, 2, \dots \quad (13)$$

де (x_m, y_m) — центр мас блоба:

$$x_m = \frac{m_{1,0}}{m_{0,0}}, \quad y_m = \frac{m_{0,1}}{m_{0,0}}. \quad (14)$$

Для виконання умови інваріантності до змін масштабу необхідно унормувати центральний момент [12]:

$$\mu_{p,q} = \frac{V_{p,q}}{m_{0,0}^{1+(p+q)/2}}. \quad (15)$$

Інваріантними до змін масштабу, обертання і переміщення є такі моменти:

$$\begin{aligned} \Phi_1 &= \mu_{2,0} + \mu_{0,2}, \\ \Phi_2 &= (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2, \\ \Phi_3 &= (\mu_{3,0} - 3\mu_{1,2})^2 + (3\mu_{2,1} - \mu_{0,3})^2, \\ \Phi_4 &= (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{2,1} + \mu_{0,3})^2, \end{aligned} \quad (16)$$

$$\Phi_5 = (\mu_{3,0} - 3\mu_{1,2})(\mu_{3,0} + \mu_{1,2}) \left[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{2,1} + \mu_{0,3})^2 \right] + \\ + (3\mu_{2,1} - \mu_{0,3})(\mu_{2,1} + \mu_{0,3}) \left[3(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2 \right],$$

$$\Phi_6 = (\mu_{2,0} - \mu_{0,2})(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{0,3} + \mu_{2,1})^2 + \\ + 4\mu_{1,1}(\mu_{3,0} + \mu_{1,2})(\mu_{0,3} + \mu_{2,1}),$$

$$\Phi_7 = 3(\mu_{2,1} + \mu_{0,3})(\mu_{3,0} - \mu_{1,2})(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{2,1} + \mu_{0,3})^2 - \\ - (\mu_{3,0} - 3\mu_{1,2})(\mu_{2,1} + \mu_{0,3}) \left[3(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2 \right].$$

Інший підхід для отримання характерних ознак із зображення об'єкта полягає в тому, що кожен виділений блоб поміщається в мінімально можливий прямокутник, який ділиться на чотири менших прямокутника Q_1, Q_2, Q_3, Q_4 , що називаються кuartилями [13]. Точкою перетину відрізків, що ділять вихідний прямокутник, є центр мас БЛОБ точка C , як це показано на Рис. 20.

Далі визначаються центри мас кuartилів q_1, q_2, q_3, q_4 та відстані d_1, d_2, d_3, d_4 до них від центру мас блоба. Для збільшення робастності характерного набору даних розраховуються також кути $\psi_1, \psi_2, \psi_3, \psi_4$, які складають вектори d_i з горизонтальним відрізком, ділить вихідний блоб. Вектор характерних ознак БЛОБ складається з восьми величин:

$$(d_1, d_2, d_3, d_4, \psi_1, \psi_2, \psi_3, \psi_4) \quad (17)$$

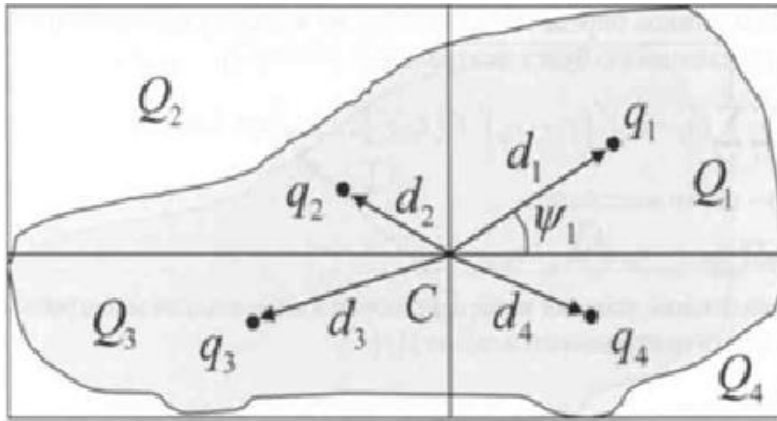


Рис. 20. Процес побудови кватилів Q_1, Q_2, Q_3, Q_4 , а також знаходження набору характерних ознак $(d_1, d_2, d_3, d_4, \psi_1, \psi_2, \psi_3, \psi_4)$. C – центр мас блоба.

Для детальної класифікації об'єктів можна скористатися методом лінійного дискримінантного аналізу. Введемо декілька класів об'єктів і обчислимо вибірку коваріаційну матрицю для точок всередині кожного класу Σ_W і між різними класами Σ_B :

$$\Sigma_W = \sum_{k=1}^K \sum_{i=1}^{n_k} (\bar{x}_{ik} - \bar{x}_k)(\bar{x}_{ik} - \bar{x}_k)^T \quad (18)$$

$$\Sigma_B = \sum_{k=1}^K n_k (\bar{x}_k - \bar{x}_0)(\bar{x}_k - \bar{x}_0). \quad (19)$$

Тут K - число класів об'єктів, n_k - кількість навчальних прикладів для класу k , \bar{x}_{ik} - вектор характерних ознак i -го навчального прикладу для класу k , \bar{x}_k - центр ваги класу k . Вектор \bar{x} - загальний центр ваги для всіх класів.

Далі знаходяться власні вектори \vec{b}_i і власні значення λ_i матриці $\Sigma_W^{-1} \Sigma_B$ за допомогою рішення рівняння $(\Sigma_B - \lambda_i \Sigma_W) \vec{b}_i = 0$. Знайдені власні значення упорядковуються за спаданням: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$, де N - розмірність простору характерних ознак. Власний вектор \vec{b}_i , що відповідає власним значенням λ_i , являє собою коефіцієнти i -ої дискримінантної функції, яка відображає вектор характерних ознак \vec{x} в точку дискримінантного простору.

Зниження розмірності можна домогтися, розглядаючи тільки $M < N$ найбільших власних значень (і, відповідно, власних векторів), таким чином відображаючи N -вимірний простір векторів характерних ознак \vec{x} в M -вимірний вектор \vec{y} :

$$\vec{y} = [\vec{b}_1 \vec{b}_2 \dots \vec{b}_M]^T \vec{x} \quad (20)$$

На практиці M вибирається з умови

$$\sum_{i=1}^M \lambda_i \geq 0.99 \sum_{i=1}^N \lambda_i \quad (21)$$

Під час класифікації визначається вектор \vec{x} для розглянутого БЛОБ і відображається в точку \vec{y} дискримінантного простору. Для визначення класу об'єкта знаходяться k найближчих сусідів точки \vec{y} , кожен з яких відповідає одному з класів. Далі клас, що найбільш часто зустрічається серед k сусідів, вважається класом даного об'єкту.

Одною з ефективних характерних ознак даного об'єкту може служити його кольорова гістограма. Позначимо через $h_i(\vec{u})$ ($\vec{u} = (R, G, B)$) гістограму об'єкта i . Тоді розподіл ймовірностей появи того чи іншого кольору об'єкта в момент часу t може бути записано таким чином:

$$P_t(\vec{u}|i) = \frac{h_i(\vec{u})}{A_i} \quad (22)$$

де A_i - розмір об'єкта.

У кожному кадрі гістограми об'єктів перераховуються за формулою (22). Після цього може бути знайдено остаточне значення ймовірності

$$P_{t+1}(\vec{u}|i) = \beta P_t(\vec{u}|i) + (1 - \beta) P_{t+1}(\vec{u}|i). \quad (23)$$

Для величини β рекомендується вибирати значення порядку 0.8.

МЕТОДИ, ЩО ЗАСНОВАНІ НА ПЕРІОДИЧНОСТІ РУХУ

Розглянемо методи, що засновані на аналізі періодичності руху, що притаманна багатьом рухомих об'єктам. Наприклад, у людини під час ходьби досить легко можна виділити пози, що повторюються через деякий час. Те ж саме відноситься і до тварин. Тому періодичність руху можна з успіхом використовувати для класифікації ряду рухомих об'єктів [15-19].

Будемо вважати рух точки $\vec{X}(t)$ в момент часу t періодичним, якщо він повторюється з постійним періодом T :

$$\vec{X}(t) = \vec{X}(t + T) + \vec{r}(t) \quad (24)$$

де $\vec{r}(t)$ - зміщення точки.

Якщо T не є константою, то рух вважається циклічним. Далі передбачається, що орієнтація і розмір об'єктів значно не змінюються протягом декількох періодів, а частота зміни кадрів досить висока, щоб вловити періодичність руху (по крайній мірі в два рази більше його частоти).

Нехай на першому етапі сегментації виділені передньопланові об'єкти F_i . Для виявлення періодичності в русі розміри кожного об'єкта необхідно привести до стандартного розміру. В іншому випадку неможливо буде врахувати зміни розмірів об'єкта при його віддаленні або наближенні до камери. Так як на етапі виділення переднього плану неминуче поява шумів, то бажано спочатку здійснити медіанну фільтрацію зображення, після чого розміри об'єктів оцінюються по декількох сусідніх кадрах. Після цього кожен об'єкт має такі атрибути: розмір, центр мас, мінімально можливий прямокутник (ММП), що обмежує об'єкт, швидкість і час існування об'єкта, що вимірюється в кадрах.

Будемо обчислювати величину самоподібності об'єкта в моменти часу t_1 та t_2 :

$$S(t_1, t_2) = \frac{\sum_{(x, y) \in \Omega} |F(x, y, t_1) - F(x, y, t_2)|}{\sum_{(x, y) \in \Omega} |F(x, y, t_1)|} \quad (25)$$

де Ω - ММП для об'єкта $F(x, y, t_1)$.

Мінімум величини S знаходиться наступним чином:

$$S'(t_1, t_2) = \min_{|dx, dy| < R} \left(\sum_{(x, y) \in \Omega} |F(x + dx, y + dy, t_1) - F(x, y, t_2)| \right) \quad (26)$$

де, наприклад, можна задати значення $R = 2$.

Для періодичного руху S' також буде періодичною. На Рис. 21 [19] представлений типовий вигляд величини S для всіх комбінацій t_1 та t_2 , що обчислений для послідовності кадрів, на яких зображена людина, що йде (темні області означають велику ступінь подібності).

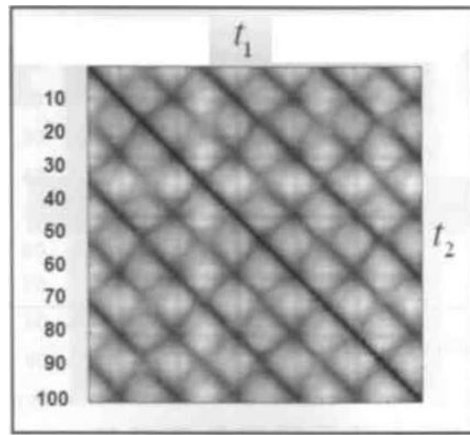


Рис. 21. Графік подібності людини, що йде

Очевидно, що графік подібності повинен бути симетричний відносно головної діагоналі. Крім того, головна діагональ завжди буде найтемнішою лінією малюнка, так як об'єкт подібний до самого себе в будь-який момент часу $S'(t, t) = 0$. Періодичний рух буде представлено темними лініями.

паралельними головної діагоналі, так як $S'\left(t, \frac{kT}{2} + t\right) \cong 0$, де T - період, а k – ціле число. Темні лінії, що перпендикулярні до головної діагоналі, утворені внаслідок того, що $S'\left(t, \frac{kT}{2} - t\right) \cong 0$ через симетрію людини, що йде (Рис. 22)

[19].

Однак не всім періодичним рухам відповідає графік S' із прямими лініями, перпендикулярні до головної діагоналі. Наприклад, для собаки, біжить, (Рис. 23 [19]) впродовж періоду не існує двох подібних кадрів (тобто, $S'(t_1, t_2) \gg 0, 0 \leq t_1 < t_2 \leq T$). Графік подібності для собаки, що біжить, наведено на Рис. 24 [19]. Ця відмінність в графіках подібності може бути використана для класифікації об'єктів.

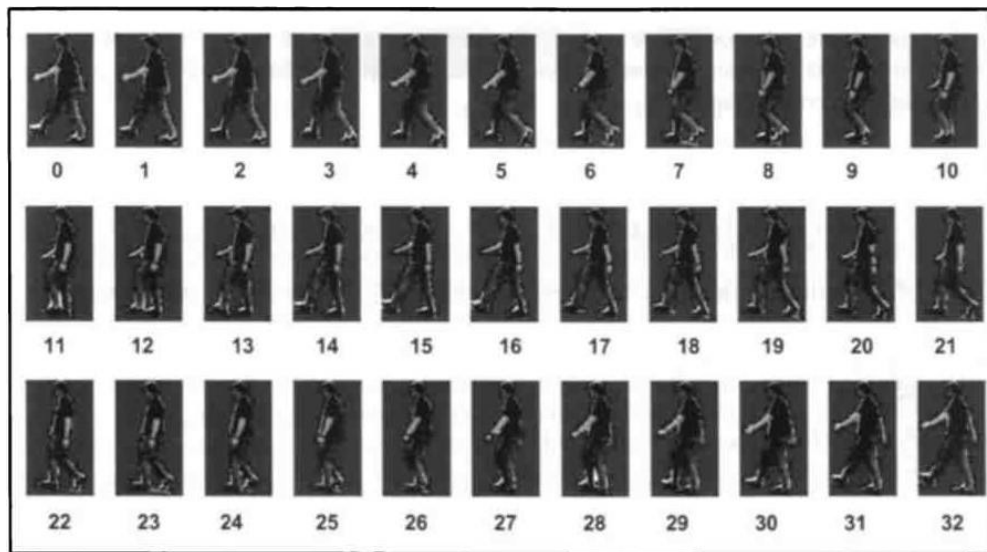


Рис. 22. Повний період людини, що йде, ($T = 32$)

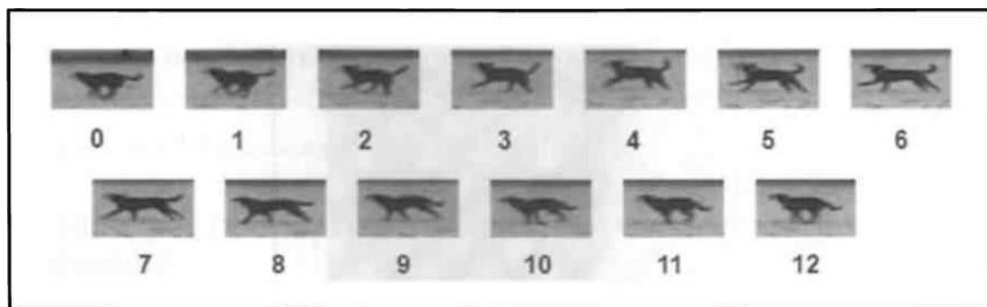


Рис. 23. Повний період собаки, що біжить, ($T = 12$)

Визначити, є рух періодичним чи ні, можна двома способами: за допомогою обчислення спектральної щільності та автокореляційної функції. У першому випадку спектральна щільність оцінюється для фіксованого

значення t_1 і всіх значень t_2 . Періодичний рух проявляється піками в спектрі щільності P . При цьому будемо вважати такий пік f_p значущим, якщо

$$P(f_p) > \mu_P + \delta \sigma_P. \quad (27)$$

де δ - порогове значення (наприклад, рівне 3), μ_P та σ_P - середнє значення і стандартне відхилення P .

У другому способі використовується A - автокореляційна функція S' . Якщо S' є періодичною величиною, то A буде мати максимуми, що рівномірно розташовані на плоскій решітці M_d , де d - відстань між вузлами решітки. Піки Θ в автокореляційній функції проєктуються на решітку M_d , за допомогою міри відповідності e :

$$e(M_d) = \sum_i |M_{d,i} - B_i|, \quad (28)$$

$$B_i = \left\{ \Theta_i \left| \begin{array}{l} |M_{d,i-\Theta_i}| \leq \min_{j \neq i} |M_{d,i-\Theta_j}| \cap |M_{d,i-\Theta_i}| \leq \delta_D \cap A(\Theta_i) < A_{\max} \end{array} \right. \right\}. \quad (29)$$

Тут B_i - найближчий пік до вузла решітки $M_{d,i}$, δ_D - порогове значення $\left(\delta_D < \frac{d}{2} \right)$, а A_{\max} - мінімальне значення автокореляційної функції, яке може приймати відповідний пік функції. Між Θ та M_d існує відповідність, якщо

$$\min_{d_1 \leq d \leq d_2} e(M_d) < \delta_e. \quad (30)$$

Тут δ_e - порогове значення відповідності, $[d_1, d_2]$ - діапазон варіації d . Для описаних величин рекомендуються наступні значення: $\delta_D = 1$, $\delta_e = 2|M_D|$, $A_{\max} = 0.25$.

Діапазон $[d_1, d_2]$ визначає можливу область зміни періоду, причому $0 < d_1 < d_2 < L$, де L - затримка автокореляційної функції.

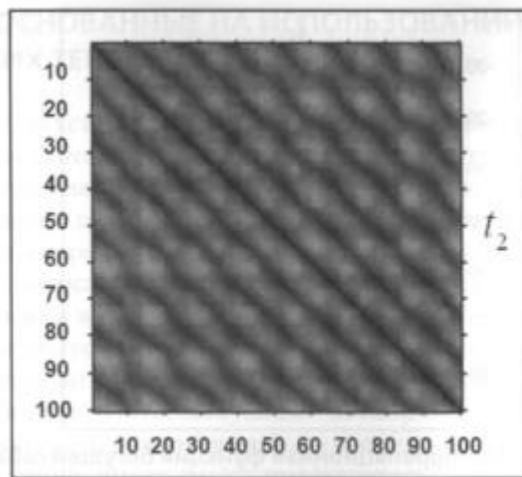


Рис. 24. Графік подібності для собаки, що біжить

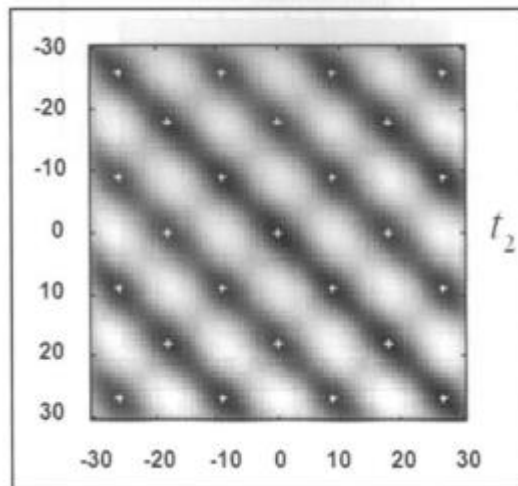


Рис. 25. Автокореляційна функція людини, що йде.

На Рис. 25 [19] зображена автокореляційна функція для відеопослідовності, що зображує людину, яка йде. Чітко видно ґратчаста структура функції A , що відображає її піки.

Автокореляційна функція для собаки, що біжить, має вигляд паралельних смуг (Рис. 26 [19]). Це є відображенням того факту, що протягом періоду для цього відеопотоку не існує двох подібних кадрів (див. Рис. 24).

Ще більш відрізняються функції подібності та автокореляційна функція автомобіля під час руху, що наведені на Рис. 27 і Рис. 28 [19].

Так як періодичності в русі транспортних засобів не спостерігається, то графік автокореляційної функції являє собою одну широку смугу, що йде уздовж головної діагоналі.

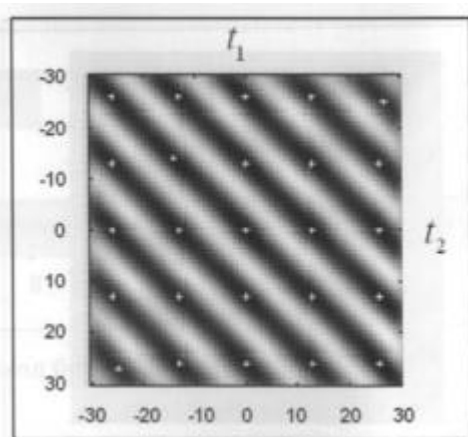


Рис. 26. Автокореляційна функція собаки, що біжить.

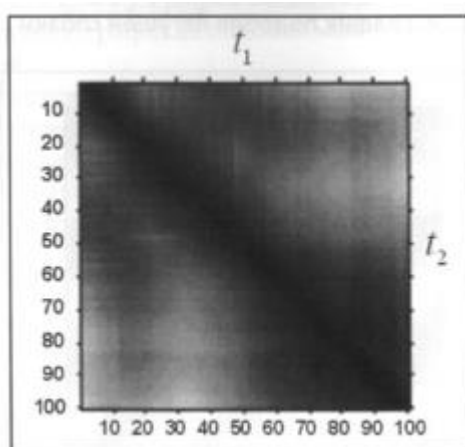


Рис. 27. Графік подібності для автомобіля, що рухається

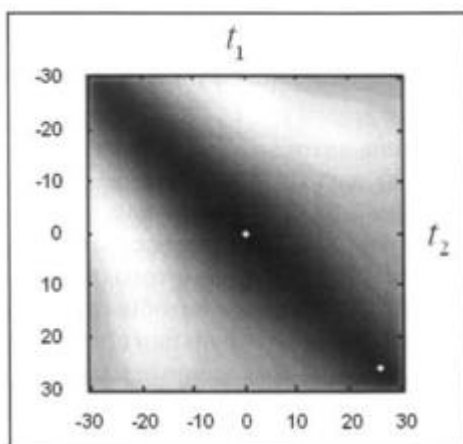


Рис. 28. Автокорреляційна функція автомобіля під час руху.

МЕТОДИ, ЩО ЗАСНОВАНІ НА ВИКОРИСТАННІ ДИНАМІЧНИХ ТЕКСТУР

Динамічні текстур - це просторово повторювані, що змінюються в часі, зорові образи, які утворюють послідовність зображень з певною часовою стійкістю. У випадку динамічних текстур (ДТ) поняття самоподібності, що є ключовим для звичайних текстур, розширюється на просторово-часову область. Типовими прикладами ДТ служать відеопослідовності, що зображують дим, вогонь, хвилі, листя, що колишеться на вітрі та ін.

Так як ДТ є тривимірним варіантом звичайних статичних текстур завдяки присутності залежності від часу, то для їх аналізу можна використовувати узагальнення відповідних двовимірних методів. Однак внаслідок того, що більшість змін в часі обумовлено наявністю руху на зображенні, має сенс заздалегідь отримати інформацію про рух. Тому природним вибором в даній ситуації є методи оптичного потоку. Основним джерелом інформації при цьому є змінне з часом векторне поле, яке представляє собою апроксимацію двовимірного поля руху, що відображає переміщення в просторі. Таке поле містить значно більше інформації, ніж скалярне поле при аналізі звичайних текстур. Більш того, напрямок і величина руху більш тісно пов'язані з конкретними подіями в порівнянні із значенням інтенсивності окремого пікселя. Це відкриває нові можливості з точки зору розпізнавання об'єктів.

Існує кілька різних підходів до аналізу динамічних текстур, основними з яких є методи оптичного потоку і методи, що використовують в якості характерних ознак ДТ параметри побудованої моделі текстури. Найбільш поширеним методом в даний час є використання нормальної компоненти оптичного потоку [20-26]. При цьому особливої важливості набуває питання коректності розгляду тільки однієї компоненти оптичного потоку. Як відомо, нормальний потік може бути ефективно обчислений і, крім того, містить корисну інформацію про рух об'єктів. Однак навіть з урахуванням його згладжування він є надзвичайно чутливим до шуму. Тісний зв'язок

нормального потоку з просторовими градієнтами проявляється в кореляції характерних ознак на його основі і зовнішніх візуальних характеристик. В цілому, поля нормального потоку не дуже добре відображають візуальну динаміку процесів. Повний оптичний потік в цьому відношенні є значно кращою характеристикою. На жаль, його обчислення в реальному часі на переважній більшості комп'ютерів не представляється можливим в даний час.

Використання оптичного потоку.

Будемо розглядати тільки нормальну складову оптичного потоку, що дозволить уникнути великого обсягу обчислень, і в той же час цієї компоненти цілком достатньо для вирішення поставлених завдань. Нормальний потік, тобто, потік в напрямку градієнта інтенсивності $\nabla I(x, y)$, записується в такий спосіб:

$$\vec{v}_N(x, y) = -\frac{dI/dt}{\|\nabla I\|} \vec{n} \quad (31)$$

де \vec{n} - одиничний вектор у напрямку $\nabla I(x, y)$. Нормальний потік містить в собі як часову, так і просторову інформацію про динамічні структури: тимчасова інформація пов'язана з рухом крайових точок, а просторова - з їх градієнтами. Недоліком застосування нормального потоку є його відносно висока чутливість до шуму. Однак вона може бути зменшена шляхом застосування згладжуючих фільтрів і подальшої еквалізації гістограм, або за допомогою порогової обробки просторових градієнтів. Крім того, якщо градієнт $\nabla I(p)$ досить рівномірно розподілений по напрямках в околиці точки $p = (x, y)$, то вираз (31) можна замінити на зважене середнє v_N що зробить його обчислення більш стійким:

$$v_0(p) = \frac{\sum_{s \in F(p)} \|\nabla I(s)\|^2 |v_N(s)|}{\max \left(\alpha, \sum_{s \in F(p)} \|\nabla I(s)\|^2 \right)} \quad (32)$$

де $F(p)$ - вікно розміром 3×3 з центром в точці p , α - задана позитивна константа, яка визначає рівень шуму в однорідних областях і запобігає діленню на нуль в деяких однорідних областях зображення, в яких градієнт дорівнює нулю. Величина v_0 являє собою міру локального руху. Пов'язана з цим втрата інформації про напрямок руху не так важлива, так як нас в першу чергу цікавить динаміка загального характеру.

Як відомо, для обчислення характерних ознак статичних текстур можуть бути використані матриці входжень, що представляють собою двовимірні масиви, в яких індекси рядків і стовпців утворюють множину допустимих на зображенні значень пікселів [27]. При цьому елементи масиву вказують, скільки разів значення першого індексу зустрічалося на зображенні в деякому заданому просторовому відношенні зі значенням другого індексу. В якості такого відношення може виступати, наприклад, сусідство пікселів.

Звичайна просторова матриця входжень може бути об'єднана з одержуваними значеннями нормальної компоненти оптичного потоку для поділу зображень з ДТ та зі звичайним рухом об'єктів [20]. Проте у такий спосіб неможливо описати часову еволюцію: досліджувані при цьому взаємодії будуть чисто просторовими і можуть характеризувати тільки стаціонарні руху. Крім того, обчислення матриць входжень з різними просторовими відносинами вимагає значних часових витрат. Тому будемо розглядати часові матриці входження, які визначаються наступним чином [21]:

$$Pd_t(i, j) = \frac{|(r, s) \in Cd_t | b(r) = i, b(s) = j|}{|Cd_t|} \quad (33)$$

де b - дискретний аналог v_0 , d_t - часовий інтервал,

$Cd_t = \{(r, s) \text{ в одній і тій же точці зображення} | \exists t : r \in I_t, s \in I_{t-d_t}\}$, а $|\Omega|$ -

позначає число елементів множини Ω .

Фактично величина (33) являє собою ймовірність появи в одній і тій же точці зображення двох величин нормальної складової швидкості, розділених часовим інтервалом d_t . Тепер ми можемо скласти ще кілька характерних ознак, похідних від (33):

$$A = \sum_{(i,j)} iP_{d_t}(i,j), \quad (34)$$

$$\sigma^2 = \sum_{(i,j)} (i-A)^2 P_{d_t}(i,j), \quad (35)$$

$$\delta = \frac{A^2}{\sigma^2}, \quad (36)$$

$$a = \sum_{(i,j)} P_{d_t}(i,j)^2, \quad (37)$$

$$c = \sum_{(i,j)} (i-j)^2 P_{d_t}(i,j). \quad (38)$$

Цей набір характерних ознак обчислюється або для заданих блоків зображення, або для областей, які є результатом просторової сегментації. За допомогою цих ознак враховується часовий аспект руху. В принципі, можна розширити описаний метод, для того щоб відобразити також і просторову складову руху за допомогою введення статистичних характеристик появи відповідних значень нормальних складових швидкості і використання багатомасштабних часових моделей Гіббса [22].

Моделі динамічних текстур

Для аналізу більшості існуючих текстур з успіхом можуть бути використані статистичні методи. Наприклад, кілька перших моментів гістограми області зображення представляють собою найпростіші характеристики текстури. Однак даний підхід має серйозний недолік: абсолютно різні текстури можуть мати практично однакові гістограми.

Модель STAR

Досить простим і в той же самий час потужним засобом аналізу є авторегресійні моделі, в яких передбачається, що кожен піксель послідовності зображень може бути представлений як лінійна комбінація сусідніх з ним точок з додатковим гаусовим шумом. Однією з таких моделей є лінійна просторово-часова авторегресійна модель (STAR) [28-29]:

$$I_t(x, y) = \sum_{i=1}^n \phi_i I_{t+\Delta t_i}(x + \Delta x, y + \Delta y) + n(x, y, t), \quad (39)$$

де $n(x, y, t)$ - білий гаусів шум, ϕ_i - параметри моделі.

Основним обмеженням моделі STAR є вимога того, щоб вхідні дані підпорядковувалися багатовимірному нормальному розподілу з постійними середнім і коваріацією. Тим самим послідовність зображень, які представляють обертання, стиснення або розширення, не можуть бути описані за допомогою цієї моделі. Більш того, статистичний характер моделі передбачає використання значної кількості пікселів для коректного аналізу текстур.

Для остаточної побудови моделі необхідно задання виду околу у виразі (39), а також способу оцінки параметрів моделі ϕ . Традиційно в аналізі часових рядів специфікація моделі проводиться шляхом обчислення автокореляційної функції і знаходження її екстремумів. Для даної тривимірної моделі STAR такий шлях є надто витратним. Тому розглянемо кілька інших підходів.

Для оцінки параметрів моделі найпростішим способом є використання тільки одного параметра, за допомогою якого кожен піксель пов'язаний з сусідніми. Даний параметр визначається шляхом наступного усереднення:

$$\phi = \frac{I_t(x, y)}{\sum_{i=1}^n I_t(x + \Delta x_i, y + \Delta y_i)} \quad (40)$$

Однак одного параметра в більшості випадків недостатньо для успішного розпізнавання динамічних текстур. Кращі результати можна отримати, якщо розбити ДТ в часі і просторі на куби розміром $m \times m \times m$ і обчислювати усереднення ϕ по кожному такому кубу. В такому випадку текстура буде характеризуватися m^3 параметрами.

Інший підхід полягає в розбитті текстурних областей $k \times k \times l$ (l - довжина відеопослідовності) на j рівних частин. Для кожної такої частини розглядається $k/2$ тривимірних околів центрального пікселя. При цьому перший параметр отримується при аналізі центрального пікселя і його найближчих сусідніх точок; другий параметр - при використанні найближчих сусідів центрального пікселя і точок, сусідніх з ними. $k/2$ параметр отримується при розгляді всіх $k/2$ околів центрального пікселя. Таким чином, для опису текстури потрібно визначити $\frac{k}{2} \times j$ параметрів.

Вище була розглянута модель першого порядку. В принципі, можна використовувати моделі більш високого порядку (MSTAR). Якщо модель STAR має на увазі, що кожен піксель текстури виражається за допомогою лінійної комбінації пікселів з деякого околу фіксованого розміру, то MSTAR є тією ж моделлю STAR, але яка визначається на різних околах, розміри яких змінюються. Так, для моделі восьмого порядку на рівні деталізації l кожен піксель представляється як

$$p_0 = \sum_{i=1}^8 p_{i,l} s_{i,l}, \quad (41)$$

де p_0 - піксель, що розглядається, $p_{i,l}$ - пікселі в кутах куба з центром в p_0 , $s_{i,l}$ - параметри, що підлягають оцінюванню.

Модель ARMA

Якщо розглядати звичайні статичні текстури, то їх можна вважати реалізацією стаціонарного стохастичного процесу з просторово

інваріантними статистиками. Для послідовності зображень окремі кадри, очевидно, не є незалежними реалізаціями стаціонарного розподілу, так як існує часова кореляція, притаманна процесу. Таким чином, розумно припустити, що окремі зображення є результатом дії динамічної системи з незалежними і однаково розподіленими процесами [30-34].

Нехай $\{I_t\}_{t=1,\dots,\tau}$, $I_t \in R^m$ є відеопослідовність, що складається з τ окремих зображень. Припустимо, що в кожен момент часу t ми маємо зашумлену версію зображення $y_t = I_t + w_t$, де $w_t \in R^m$ і описується розподілом $p_W(\bullet)$, а $y_t \in R^m$, $t = 1, \dots, \tau$.

Тоді будемо вважати, що послідовність $\{I_t\}$ являє собою лінійну динамічну текстуру, якщо існує набір з n фільтрів $\varphi_\alpha : R \rightarrow R^m$, $\alpha = 1, \dots, n$ і стаціонарний розподіл q такий, що для $x_t \in R^n$ і $I_t = \varphi(x_t)$ отримаємо

$$x_t = \sum_{i=1}^k A_i x_{t-i} + B v_t, \quad (42)$$

де $v_t \in R^n$ - незалежні і однаково розподілені величини з щільністю $q(\bullet)$, а $A_i \in R^{n \times n}$, $i = 1, \dots, k$, $B \in R^{n \times n}$ і $x_0 = \bar{x}_0$.

Без втрати загальності будемо вважати, що $k = 1$. Тоді динамічні текстури описуються за допомогою авторегресійної моделі козаючого середнього (ARMA) з невідомим вхідним розподілом:

$$\begin{cases} x_{t+1} = A x_t + B v_t \\ y_t = \varphi(x_t) + w_t \end{cases}, \quad (43)$$

де $x_0 = \bar{x}_0$, $v_t \sim q(\bullet)$ і $w_t \sim p_W(\bullet)$ - незалежні і однаково розподілені, v_t - невідомі, w_t задані так, що $I_t = \varphi(x_t)$.

Очевидно, що для подальшого аналізу нам необхідно задати вид фільтрів φ_α . Це може бути зроблено декількома способами.

Найпростіший з них полягає в розгляді динаміки окремих пікселів $x_t = I_t$. Але більш адекватним є наступний вибір:

$$I_t = \sum_{i=1}^n (x_t)_i \theta_i = Cx_t, \quad (44)$$

де $C = [\theta_1, \dots, \theta_n] \in R^{n \times m}$, а $\{\theta_i\}$ - являє собою ортонормованій базис в просторі L_2 або ж набір головних компонент. Можлива також нелінійна форма фільтрів.

Маючи послідовність зашумлених зображень $\{y_t\}_{t=1, \dots, \tau}$, опис динамічної текстури зводиться до завдання параметрів моделі A, B, C і щільності розподілу $q(\bullet)$ в (43). Це так звана задача системної ідентифікації [32], коли необхідно побудувати динамічну модель за допомогою часових рядів. Однак на відміну від розглянутого випадку при вивченні динамічних систем розподіл $q(\bullet)$ зазвичай відомо. Таким чином, завдання опису динамічних текстур виглядає наступним чином:

для заданих y_1, \dots, y_τ знайти

$$\hat{A}, \hat{B}, \hat{C}, \hat{q}(\bullet) = \arg \max_{A, B, C, q} \log p(y_1, \dots, y_\tau), \quad (45)$$

при виконанні (43) і умови, що v_t - незалежні однаково розподілені із щільністю $q(\bullet)$ величини.

Як відомо, стаціонарний процес другого порядку з довільною коваріацією може бути представлений як результат дії динамічної системи, що описується гауссовим розподілом з нульвим середнім [32]. Тому будемо вважати, що існує додатне ціле n , процес $x_t \in R^n$ з початковою умовою $x_0 = \bar{x}_0 \in R^n$ і симетричні додатньо визначені матриці $Q \in R^{n \times n}$, $G \in R^{m \times m}$ такі, що

$$\begin{cases} x_{t+1} = Ax_t + v_t, & v_t \sim \eta(0, Q), & x_0 = \bar{x}_0 \\ y_t = Cx_t + w_t, & w_t \sim \eta(0, G) \end{cases}, \quad (46)$$

для деяких матриц $A \in R^{n \times n}$ і $C \in R^{m \times n}$. Тут $\eta(\mu, s)$ — нормальний розподіл із середнім μ та дисперсією s . Задача системної ідентифікації полягає в оцінці параметрів моделі A, C, Q, G за допомогою даних y_1, \dots, y_t .

Замітимо, що B і v_t в моделі (43) задовільняють умовам $BB^T = Q$ і $v_t \sim \eta(0, E_{n_v})$, де E_{n_v} — одинична матриця розміром $n_v \times n_v$.

З (46) випливає, що для отримання однієї і тієї ж послідовності $\{y_t\}$ матриці A, C і Q можна вибрати не єдиним способом. Іншими словами, кожен процес описується не однією моделлю, а цілим класом еквівалентних моделей. Тому нам необхідно накласти додаткові умови для отримання єдиної канонічної моделі. Наприклад, це можуть бути наступні припущення:

$$m \gg n; \text{rang}(C) = n, \quad (47)$$

$$C^T C = E_n, \quad (48)$$

де E_n - одинична матриця розміром $n \times n$.

Тепер завдання формулюється в такий спосіб:

для даних y_1, \dots, y_τ , $\tau \gg n$ знайти $\hat{A}, \hat{B}, \hat{C}, \hat{G}$ - канонічну модель для процесу $\{y_t\}$:

$$\hat{A}_\tau, \hat{C}_\tau, \hat{Q}_\tau, \hat{G}_\tau = \arg \max_{A, C, Q, G} p(y_1, \dots, y_\tau). \quad (49)$$

Існує асимптотично оптимальне рішення цієї задачі, проте його отримання пред'являє надзвичайно жорсткі вимоги до обчислювальних ресурсів. Тому ми розглянемо субоптимальне рішення, яке значно швидше з точки зору його чисельної реалізації.

$$\text{Позначимо } Y_1^\tau = [y_1, \dots, y_\tau] \in R^{m \times \tau},$$

$$X_1^\tau = [x_1, \dots, x_\tau] \in R^{n \times \tau}, \quad \tau > n$$

$$W_1^\tau = [w_1, \dots, w_\tau] \in R^{m \times \tau}.$$

Тоді

$$Y_1^\tau = CX_1^\tau + W_1^\tau, \quad C \in R^{m \times n}, \quad C^T C = E_n.,$$

Припустимо, що Y_1^τ у, являє собою SVD-розклад, де $Y_1^\tau = U\Sigma V^T$,

$U \in R^{m \times n}$, $U^T U = E_n$, $V \in R^{\tau \times n}$, $V^T V = E_n$, Σ -діагональна матриця

$$\Sigma = \begin{pmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_n \end{pmatrix} \quad (51)$$

Будемо шукати C і X такі, що $\hat{C}_\tau, \hat{X}_\tau = \arg \min_{C, X_1^\tau} \|W_1^\tau\|_F$ за умови (50).

Тут норма розуміється в сенсі Фробеніуса

$$\|W\|_F = \sqrt{\sum_i \sum_j |w_{ij}|^2}. \quad (52)$$

Тоді єдине рішення визначається наступним чином:

$$\hat{C}_\tau = U, \quad \hat{X}_\tau = \Sigma V^T, \quad (53)$$

а матриця \hat{A} може бути визначена єдиним чином з рішення наступної лінійної задачі:

$$\hat{A}_\tau = \arg \min_A \|X_1^\tau - Ax_0^{\tau-1}\|_F. \quad (54)$$

Взаховуючи (52), отримаємо

$$\hat{A}_\tau = \Sigma V^T D_1 V \left(V^T D_2 V \right)^{-1} \Sigma^{-1}, \quad (55)$$

де

$$D_1 = \begin{pmatrix} 0 & 0 \\ E_{\tau-1} & 0 \end{pmatrix}, \quad D_2 = \begin{pmatrix} E_{\tau-1} & 0 \\ 0 & 0 \end{pmatrix} \quad (56)$$

Величина Q може бути знайдена з наступного виразу:

$$\hat{Q}_\tau = \frac{1}{\tau} \sum_{i=1}^{\tau} \hat{v}_i \hat{v}_i^T, \quad (57)$$

де

$$\hat{v}_t = x_{t+1} - \hat{A}_\tau \hat{x}_t. \quad (58)$$

У розглянутому алгоритмі значення n передбачалося заданим. Однак порядок моделі можна визначити за допомогою значень $\sigma_1, \sigma_2, \dots$, вибравши n таким чином, щоб всі величини σ_j не перевищували якогось порогового значення, яке може, наприклад, залежати від різниці σ_j і σ_{j-1} .

Знайшовши параметри моделі, можна проводити сегментацію зображення, для чого необхідно встановити міру відповідності між областями. Складність полягає в тому, що ці області описуються не просто параметрами моделі, а цілим класом еквівалентних параметрів, які можуть бути отримані шляхом зміни базису змінних стану x_t , в моделі (43). Тому адекватна міра відповідності повинна порівнювати між собою не параметри, а їх еквівалентні класи.

Одним з можливих шляхів вирішення даної проблеми є визначення кутів між багатовимірними підпросторами простору всіх можливих зображень.

Нехай $M(1)$ і $M(2)$ є дві моделі (43) однієї і тієї ж розмірності, які характеризуються відповідно системними матрицями $A(1)$ і $A(2)$, і вихідними матрицями $C(1)$ і $C(2)$.

Введемо наступні матриці:

$$O(i) = \left[C(i)^T \quad A(i)^T C(i)^T \dots \left(A(i)^T \right)^n C(i)^T \dots \right]^T \in R^{\infty \times n}, \quad i=1,2 \quad (59)$$

і визначимо наступні кути:

$$\cos(\theta_1) = \frac{\left| x_{(1)}^T O_{(1)}^T O_{(2)} y_{(1)} \right|}{\|O_{(1)} x_{(1)}\| \cdot \|O_{(2)} y_{(1)}\|}, \quad (60)$$

$$\cos(\theta_k) = \frac{\left| x_{(k)}^T O_{(1)}^T O_{(2)} y_{(k)} \right|}{\|O_{(1)} x_{(k)}\| \cdot \|O_{(2)} y_{(k)}\|}, \quad k = 2, \dots, n \quad (61)$$

за умови $x_{(i)}^T O_{(1)}^T O_{(1)} x = 0$ і $y_{(i)}^T O_{(2)}^T O_{(2)} y = 0$ для $i = 1, 2, \dots, k - 1$.

Тоді відповідність різних моделей визначається як

$$d_m^2(M_1, M_2) = \ln \left(\prod_{k=1}^n \frac{1}{\cos^2(\theta_k)} \right). \quad (62)$$

ЛІТЕРАТУРА

1. A. P Pentland. Classification By Clustering. In: PTC. Symposium on Machine Processing of Remotely Sensed Data, Purdue, 1976.
2. R. J. Kauth, A. P Pentland, C. S nomas. Blob: An Unsupervised Clustering Approach to Spatial Preprocessing of Mass Imagery. In: PTC. International Symposium on Remote Sensing of the Environment, Ann Arbor, p. 1309-1317, 1977.
3. D. Zhang, G. Lu. Segmentation of Moving Objects in Image Sequence: A Review. In: Circuits, Systems and Signal Processing, v. 20, p. 143-183, 2001.
4. P Salembier, J Serra. Flat Zones Filtering, Connected Operators, and Filters by Reconstruction. In: IEEE Transactions on Image Processing, v.4, p. 1153-1160, 1995.
5. A. K. Jain, R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, New Jersey, 1988, 320 p.
6. H. J. A. M. Heijmans. Connected Morphological Operators for Binary Images. In: Computer Vision and Image Understanding, v 73. p. 99-120, 1999.
7. U. Braga-Neto, J Goutsias. Automatic Target Detection and Tacking in Forwardlooking Image Sequences Using Morphological Connected Operators. In: Proc Conference on Information Sciences and Systems, Baltimore, v. 1, 173-178, 1999.
8. S Batman, N. Osman, J. Prince, J Goutsias. Automatic Myocardial Segmentation of Tagged MR Cardiac Images Using Morphology. In: PTC. Conference on Information Sciences and Systems, Princeton, v. 1, p. 65-70, 2000.
9. M. Piccardi (October 2004). Background subtraction techniques: a review (PDF). IEEE International Conference on Systems, Man and Cybernetics. 4. pp. 3099–3104. doi:10.1109/icsmc.2004.1400815.
10. B. Tamersoy (September 29, 2009). "Background Subtraction – Lecture Notes" (PDF). University of Texas at Austin.

11. B. Patel; N. Patel (March 2012). Motion Detection based on multi-frame video under surveillance systems. Vol. 12.
12. N. Lu; J. Wang; Q. Wu; L. Yang (February 2012). An improved Motion Detection method for real time Surveillance.
13. Y. Benezeth; B. Emile; H. Laurent; C. Rosenberger (December 2008). Review and evaluation of commonly-implemented background subtraction algorithms (PDF). International Conference on Pattern Recognition. pp. 1–4. doi:10.1109/ICPR.2008.4760998.
14. C. Wren; A. Azarbayejani; T. Darrell; A. Pentland (July 1997). "Pfinder: real-time tracking of the human body" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 19 (7): 780–785. doi:10.1109/34.598236. hdl:1721.1/10652. Archived from the original (PDF) on 2007-06-09.
15. C. Stauffer; W. Grimson (August 1999). Adaptive background mixture models for real-time tracking (PDF). IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2. pp. 246–252. doi:10.1109/CVPR.1999.784637.
16. T. Bouwmans; F. El Baf; B. Vachon (November 2008). "Background Modeling using Mixture of Gaussians for Foreground Detection – A Survey". Recent Patents on Computer Science. 1: 219–237.
17. P. Power; J. Schoonees (2002). "Understanding Background Mixture Models for Foreground Segmentation" (PDF). Proceedings Image and Vision Computing New Zealand 2002: 267–271.
18. T. Bouwmans (November 2009). "Subspace Learning for Background Modeling: A Survey". Recent Patents on Computer Science. 2 (3): 223–234. doi:10.2174/1874479610902030223.
19. T. Bouwmans (January 2010). Statistical Background Modeling for Foreground Detection: A Survey. Chapter 3 in the Handbook of Pattern Recognition and Computer Vision, World Scientific Publishing. pp. 181–199. doi:10.1142/7297. ISBN 978-981-4273-38-1.

