

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
« ____ » _____ 20__ р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА
ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

Інформаційна система детектору рухомих об'єктів

Освітній ступінь “бакалавр”
Спеціальність 123 – “комп'ютерна інженерія”

Керівник проекту:

(підпис)

Рязанцев О.І.

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Критська Я.О.

(ініціали, прізвище)

Здобувач вищої освіти:

(підпис)

Джафаров В.В.

(ініціали, прізвище)

Група:

КІ-15з

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Комп'ютерних наук та інженерії

Освітній ступінь бакалавр

Напрямок підготовки _____

(шифр і назва)

Спеціальність 123 – комп'ютерна інженерія

(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри КНІ

І.С. Скарга-Бандурова

« _____ » _____ 20 ____ р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Джафарову В'ячеславу Вахідовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна система детектору рухомих об'єктів

керівник проекту (роботи) Рязанцев Олександр Іванович, д.т.н., проф.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "13" 05 2019 р. № 84/15.15

2. Термін подання студентом роботи 16.06.2019

3. Вихідні дані до роботи Витяги з робіт вчених які займалися схожими

проблемами, опис відомих методів, системне програмне забезпечення: ОС

Windows 7/8/10, мередовище програмування: Intelij IDEA 2018 Ultimate Edition

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно

розробити) Аналіз фіксації руху та постановка задачі, моделювання даних для

розпізнавання руху, опис програмної реалізації проекту, охорона праці

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст. викл. Критська Я.О.		

7. Дата видачі завдання 30.04.2019

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз завдання та робота з літературою	05.05.2019 - 13.05.2019	
2	Аналіз технічних засобів	14.05.2019 - 22.05.2019	
3	Розробка алгоритму	22.05.2019 - 02.06.2019	
4	Програмна реалізація	02.06 .2019- 11.06.2019	
5	Оформлення пояснювальної записки та електронних плакатів	11.06.2019 - 16.06.2019	

Здобувач вищої освіти

_____ (підпис)

Джафаров В.В.

_____ (прізвище та ініціали)

Керівник

_____ (підпис)

Рязанцев О.І.

_____ (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 76 с., 17 рис., 3 табл., 1 додаток, 25 джерел.

Робота присвячена вирішенню проблеми розпізнавання руху об'єктів у відеоряді. В результаті роботи було розроблено алгоритм, що реалізує ряд методів обробки відеозображень і виявлення руху в потоці відеоданих. Створено детектор рухомих об'єктів, що використовує розроблений алгоритм.

Програма дозволяє відокремлювати динамічну складову в відео потоці. На основі аналізу даних відеоряду відбувається пошук рухомих об'єктів. Система розпізнавання руху, що покладено в основу програми, базується на власному алгоритмі, який порівнює два порядкових кадра відеоряду та знаходить рухомі об'єкти, які задовольняють встановленим налаштуванням.

Програмний комплекс в реальному часі оброблює відео з камери спостереження у будинку. Розраховує низку характеристик геометричних та динамічних параметрів об'єкту, що рухається (наприклад, габарити, швидкість та інші). Результат системи розпізнавання дозволяє обрати контролюючу дію, наприклад надіслати власнику будівлі застереження за допомогою SMS повідомлення або електронної пошти.

Запропонований охоронний програмний комплекс полегшить спостереження за будинком, зробить його більш ефективним за допомогою відправки повідомлень користувачу та дозволить заощадити на покупці детекторів руху, через можливість використовувати будь-яку відеокамеру.

Ключові слова: ВІДЕОРЯД, ФІЛЬТРАЦІЯ, РАСТР, МІЖКАДРОВА РІЗНИЦЯ, ПЕРЕТВОРЕННЯ ЗОБРАЖЕНЬ, ВИЯВЛЕННЯ РУХУ, OPENCV, JAVA.

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А., м. Сєверодонецьк, 93400с.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ФІКСАЦІЇ РУХУ ТА ПОСТАНОВКА ЗАДАЧІ	8
1.1 Актуальність проблеми розпізнавання руху на відеоряді.....	8
1.2 Опис предметної області.....	9
1.3 Складність алгоритмів	12
1.4 Постановка задачі.....	14
2 МОДЕЛЮВАННЯ ДАНИХ ДЛЯ РОЗПІЗНАВАННЯ РУХУ	15
2.1 Предобробка растрових даних.....	15
2.1.1 Перешкоди при реєстрації зображень	15
2.1.2 Типи операцій	16
2.1.3 Зменшення кадрового розміру	18
2.1.4 Лінійна фільтрація	19
2.1.5 Медіанна фільтрація.....	22
2.1.6 Помірна фільтрація.....	23
2.1.7 Міжкадрова фільтрація.....	25
2.1.8 Коригування яскравості.....	25
2.1.9 Коригування контрастності.....	27
2.2 Виявлення руху.....	29
2.3 Міжкадрова різниця	30
2.3.1 Метод базового кадру.....	32
2.3.2 Математична морфологія	34
2.3.3 Виділення об'єктів.....	35
2.3.4 Рекурсивний пошук	37
2.3.5 Пошук хвильовим методом.....	38
2.3.6 Трасування	39
2.3.7 Фільтрація об'єктів за їх параметрами.....	40
2.4 Розробка алгоритму фіксації руху	42
2.4.1 Предобробка.....	43

	5
2.4.2 Фіксація руху	44
3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ПРОЕКТУ	48
3.1 Огляд мови JAVA.....	48
3.2 Огляд бібліотеки OpenCV	49
3.3 Програмна реалізація	50
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.	54
4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проектowanego об'єкту, що мають вплив на персонал	54
4.2 Заходи щодо техніки безпеки	55
4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці	58
4.4 Рекомендації по пожежній безпеці.....	62
ВИСНОВКИ	67
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	68
ДОДАТОК 1. Електронні плакати.....	71

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

СТЗ – система технічного зору

ДР – датчик руху

ВСТУП

В даний час людство виразно рухається в бік наростання комфорту в наших оселях: сучасна квартира дуже часто крім безлічі побутової техніки та аудіо-відео комплексу містить ще й системи кондиціонування, опалення, освітлення і охорони. Під час дослідження було вирішено загострити увагу саме на реалізації останньої. Актуальність роботи підтверджується статистикою зросту інтересу до розумних технологій та можливостей автоматизації повсякденних дій.

Розпізнавання образів є областю, яка використовує різноманітні методи для отримання інформації з відеоданих, що, в основному, базуються на статистичному підході. У цій галузі розроблено багато різноманітних програм, бо задача спостереження за динамічними об'єктами є досить актуальною. Спостереження включає моніторинг, розпізнавання руху та прийняття рішень. Саме цій низці проблем і присвячено цю роботу.

Протягом останнього десятиліття було розроблено складні алгоритми аналізу відео: підрахунок відвідувачів, підрахунок конверсії, статистика касових операцій і багато іншого. У цій системі зникає оператор спостереження – ми залишаємо комп'ютеру можливість аналізувати та робити висновки.

Очевидний приклад розумного відеоспостереження – детектор руху. Один детектор руху здатний замінити відразу кілька операторів відеоспостереження.

Пропонується запровадити охоронний програмний комплекс стеження, який дозволить використовувати будь яку камеру в якості детектору руху. В такому разі буде неважливо чи є вбудований детектор в самій камері – буде використовуватися детектор програми.

1 АНАЛІЗ ФІКСАЦІЇ РУХУ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Актуальність проблеми розпізнавання руху на відеоряді

В даний час людство виразно рухається в бік наростання комфорту в наших оселях: сучасна квартира дуже часто крім безлічі побутової техніки та аудіо-відео комплексу містить ще й системи кондиціонування, опалення, освітлення і охорони. Під час дослідження було вирішено загострити увагу саме на реалізації останньої. Актуальність роботи підтверджується статистикою зросту інтересу до розумних технологій та можливостей автоматизації повсякденних дій.

Особлива увага була надана вивченню інтеграції відеоспостереження з цифровими системами, яка відкрила можливість автоматизовано отримувати різні дані, аналізуючи послідовність зображень. Важливість важко переоцінити: в звичайному випадку після 12 хвилин безперервного спостереження оператор починає пропускати до 45% подій. І до 95% потенційно тривожних подій буде пропущено вже після 22 хвилин безперервного спостереження (за результатами дослідження IMS Research, 2002)

Протягом останнього десятиліття було розроблено складні алгоритми аналізу відео: підрахунок відвідувачів, підрахунок конверсії, статистика касових операцій і багато іншого. У цій системі зникає оператор спостереження – ми залишаємо комп'ютеру можливість аналізувати та робити висновки.

Очевидний приклад розумного відеоспостереження – детекція руху. Один детектор руху здатний замінити відразу кілька операторів відеоспостереження.

1.2 Опис предметної області

Під відеоданими зазвичай розуміється потік даних, що мають структуру послідовних кадрів відеоряду. Ці кадри можуть бути спочатку отримані від пристрою реєстрації (цифрової / аналогової камери) або синтезовані за допомогою спеціального програмного забезпечення, а потім неодноразово перетворені для зручності їх зберігання і передачі. Всі кадри подібного потоку мають один розмір зображення і зазвичай надходять з певним постійним тимчасовим інтервалом. Величина часового інтервалу між сусідніми кадрами, – це кількість кадрів, яку необхідно обробити за одну секунду (frames per second або fps). Крім того, всі кадри потоку мають однакову глибину кольору, вимірювану в кількості використаних бітів пам'яті для зберігання одного пікселя кадру (bits per pixel або bpp). Зауважимо, що ми будемо мати справу виключно з цифровими відеоданими, тобто, з тими, які зберігаються в пам'яті ЕОМ і обробляються на ній програмно або апаратно.

Обробка даних відеоряду часто полягає в [3]:

- реєстрації кадрів відеозображення;
- кодування / декодування растрових даних;
- незворотному перетворенні растрових даних, що включає точкові, локальні або глобальні перетворення пікселів кадрів;
- пошуку певних елементів зображення, обчислення характеристик зображення.

Залежно від поставленого завдання, над потоком відеоданих може послідовно виконуватися одна або кілька подібних операцій. Реєстрація (відеозахват) зазвичай відбувається апаратно за допомогою відеокамери, інші етапи частіше реалізуються програмно. Кодування / декодування може виконуватися різними кодеками, що реалізують алгоритми безвтратного стиснення растрових даних або стиснення з втратами. Решта перетворень можуть включати такі операції, як обробка фільтрами згладжування,

виділення країв, зміни контрастності, яскравості, колірному простору та ін. Перетворювачі, що виконують подібні дії будемо називати фільтрами, оскільки їхні уділи програмного коду, реалізують не що інше, як алгоритми фільтрації растрів.

Фільтр детектора руху також виконує певні перетворення над растровими даними, щоб виділити необхідні елементи зображення і придушити інші. Після чого елементи реєструються в якості рухомих об'єктів і далі можуть або класифікуватися, або зберігатися в базі даних об'єктів, що рухаються, або відображатися на екрані дисплея оператора в разі, якщо система не повністю автоматизована і вимагає прийняття рішення людиною. Таким чином, ланцюжок обробки растрових даних детектором руху зазвичай будується з кількох фільтрів перетворення даних (предоброки) і власне детектора рухомих об'єктів.

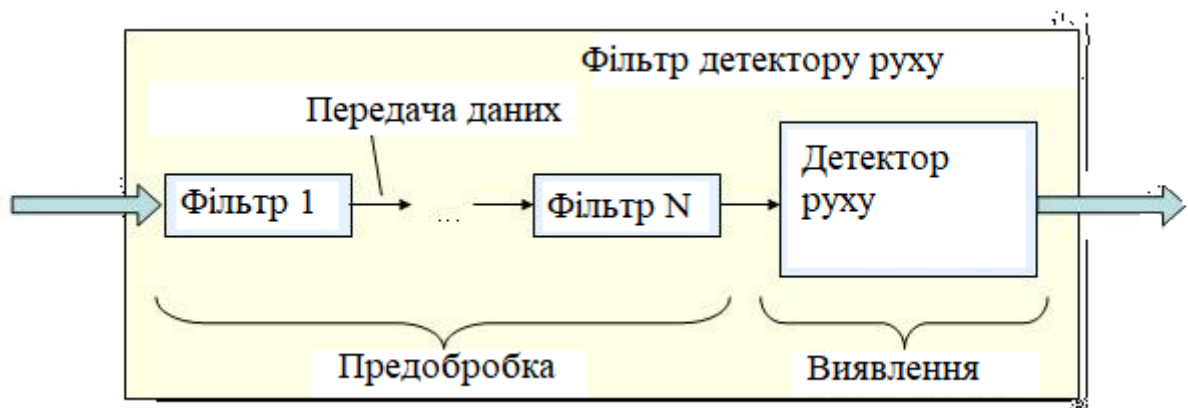


Рисунок 1.1 – Внутрішня структура фільтра детектора руху

За принципом роботи детектори руху можна розбити на три класи:

- різницеві (використовують попіксельне обчислення різниць);
- контурні (засновані на виявленні контурів об'єктів);
- інші (в їх основі лежать інші принципи, наприклад, вейвлет перетворення [4]).

У даній роботі розглядається клас різницевих детекторів з кількох причин:

- низькі часові витрати на етапі попередньої обробки растрових даних. Експерименти показали, що досить продуктивний процесор (використовувався процесор Intel Pentium 4 з тактовою частотою 2.8GHz) може обробляти до 4 каналів відео з роздільною здатністю 640x480 пікселів і глибиною кольору 24 біта на швидкості до 15 кадрів в секунду);
- зручність оптимізації за допомогою SIMD-інструкцій процесора;
- можливість виявлення швидких і повільних рухомих об'єктів, а також зайвих предметів;
- наявність досвіду розробки подібних алгоритмів.

Всі алгоритми, які будуть згадуватися далі, будемо розглядати стосовно колірному простору RGB при глибині кольору 24 біта. Для зручності в окремих випадках ми будемо використовувати 32-бітові пікселі, зокрема, коли буде необхідно обробляти піксель не покомпонентно, а цілком за одну операцію. У подібних випадках зручно працювати не з 24 bpp, а з 32 bpp. Зазвичай при роботі в режимі RGB32 четвертий байт використовується для зберігання інформації про прозорість пікселя (альфа-каналу), але ми будемо залишати його порожнім. Також необхідно зауважити, що колір в форматі RGB в пам'яті ЕОМ, насправді, зберігається в зворотному порядку, тому при безпосередньому читанні і запису пікселя в растр отримуємо BGR для 24-бітного пікселя і XBGR для 32-бітного, де, для У нашій випадку, X - порожня компонента (рис.3), яку ми будемо заповнювати нулем.

Ми не будемо торкатися 16-бітні режими RGB типу RGB555 і RGB565 з трьох причин. По-перше, тому, що робота з ними менш зручна і наочна, оскільки для виокремлення компоненти кольору пікселя необхідно обробляти його біти за допомогою масок і зрушень. По-друге, алгоритми перетворення з 16-бітного RGB-простору в 24-бітове не мають складнощів у реалізації й існують у вигляді стандартних OpenCV-фільтрів, що поставляються разом з операційною системою Windows. Нарешті, практично

будь-яка цифрова камера надає користувачеві можливість вибрати між RGB16 і RGB24.

Робота з зображеннями у відтінках сірого (8 bpp, grayscale) є окремим випадком роботи з форматом RGB24, оскільки повністю аналогічна обробці будь-якої однієї з компонент 24-бітного кольору. Крім того, при роботі з кольором, на відміну від обробки відтінків сірого, доводиться вирішувати додатково завдання об'єднання результатів, по-отриманих для кожної з трьох компонент. Тому багато уваги зображень у відтінках сірого в даній роботі не приділяється.

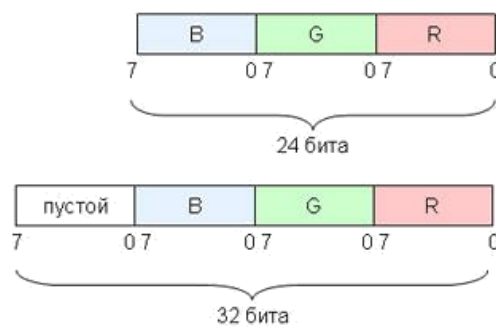


Рисунок 1.2 – RGB24 і RGB32 з незадіяним байтом альфа-каналу

1.3 Складність алгоритмів

Спробуємо визначити максимальну складність алгоритмів, які ми можемо використовувати, щоб обробка ними наших відеоданих залишалася в межах реального часу.

Обсяг інформації, що надходить на вхід алгоритму за секунду, в загальному випадку можна розрахувати за формулою (1.1)

$$N = f \cdot w \cdot h \cdot d \quad (1.1)$$

де f — частота зйомки камерою (Гц), w — ширина растра (пікс.), h — висота растра (пікс.), d — глибина кольору або число байт на один піксель (байт), N — загальний обсяг інформації на вході алгоритму за секунду (байт).

Візьмемо середні умови: нехай на вхід алгоритму надходять дані від камери. Камера працює на частоті 15 Гц, розмір зображення: 640 на 480 пікселів, глибина кольору 3 байта на піксель. Разом за секунду алгоритм повинен встигнути обробити 13824Кб вхідних даних.

Нехай наш процесор працює на частоті 2ГГц, тобто робить 2×10^9 тактів за секунду, таким чином, за формулою (1.2)

$$n = F/N, \quad (1.2)$$

де n — число тактів процесора на обробку одного байта даних за одну секунду, F - тактова частота процесора (тактів в секунду), на обробку одного байта вхідних даних за секунду припадатиме в середньому 144 такту процесора. Це не так багато, якщо також враховувати, що більшість інструкцій CISC-процесора, якими є практично всі процесори, нині випускаються фірмами Intel і AMD2, виконується далеко не за один такт, а для обробки растрових даних також необхідні операції читання / запису в пам'ять.

Щоб на один байт вхідних даних припадав хоча б один такт процесора за секунду ($n = 1$), тобто, щоб наші розрахунки вкладалися в реальний час, обсяг вхідних даних, за формулою (1.3)

$$N = F/n, \quad (1.3)$$

повинен обмежуватися максимум 2×10^9 байтами в секунду.

Якщо при наших середніх умовах на вхід надходить 13824Кб даних в секунду, то алгоритму з трудомісткістю порядку $O(n)$ знадобиться $\approx 13.8 \times 10^6$ тактів процесора для розрахунків, а алгоритму з трудомісткістю порядку $O(n^2)$ знадобиться вже $\approx 1,9 \times 10^{14}$ тактів процесора, що відразу на 5 порядків перевищує допустиму величину.

Таким чином, для вирішення поставленого завдання має сенс розглядати тільки алгоритми, що мають трудомісткість, меншу, ніж $O(n^2)$. Алгоритми, що мають велику трудомісткість, можна відразу виключити з розгляду, бо вони не задовольняють умовам задачі.

1.4 Постановка задачі

Розробка детектора руху полягає в дослідженні і застосуванні методів обробки (фільтрації) растрових зображень і виявлення на них рухомих об'єктів, таким чином, щоб забезпечити виконання наступних вимог:

- можливість роботи детектора в реальному часі;
- стійку роботу в широкому діапазоні зовнішніх умов;
- мінімальна кількість помилкових спрацьовувань на шум;
- мінімальна кількість спрацьовувань на об'єкти, які не становлять небезпечності для охоронюваного периметра / об'єкта (хитні дерева, листя, інша рослинність та ін.).

Якість роботи розробленого детектора перевіряється на відеороликах, знятих різними цифровими відеокамерами в широкому діапазоні зовнішніх умов.

2 МОДЕЛЮВАННЯ ДАНИХ ДЛЯ РОЗПІЗНАВАННЯ РУХУ

2.1 Предобробка растрових даних

Основним завданням етапу предобробки растрових даних є підготовка зображень до виявлення на них рухомих об'єктів. Тому попередня обробка повинна зводитися до виділення на зображенні тих його елементів, які, ймовірно, належать рухомим об'єктам. Поки нам досить вважати, що рухається те, що людина сприймає, як об'єкт, що рухається, спостерігаючи оком один кадр відеоряду за іншим. Далі ми дамо більш точне формулювання рухомого об'єкта, з точки зору детектора руху.

Оскільки ми зупинилися на різницевому підході до виявлення руху, заснованого на припущенні про те, що нерухомий фон відчуває слабкі зміни форми і кольору від кадру до кадру, то необхідно домогтися, щоб ці зміни були дійсно мінімальні. Для цього достатньо відфільтрувати шуми і мінімізувати зміни рівня яскравості і контрастності між сусідніми кадрами.

2.1.1 Перешкоди при реєстрації зображень

Ніяка система реєстрації не забезпечує ідеальної якості зображень досліджуваних об'єктів. Зображення в процесі їх формування реєструються пристроями (фотокамерами, відеокамерами, датчиками температури та ін.) Зазвичай піддаються впливу різних випадкових перешкод або шумів. На відміну від так званих детермінованих спотворень, які часто описуються поелементно функціональними перетвореннями вихідного зображення, для опису випадкових впливів використовують моделі адитивної, імпульсного і мультиплікативного шумів [5].

Найбільш поширеним видом перешкод є випадковий адитивний шум, статистично незалежний від відеосигналу. Модель адитивного шуму використовується тоді, коли сигнал на виході або на якомусь проміжному етапі перетворення може розглядатися як сума корисного сигналу і деякого

випадкового сигналу (шуму). Модель адитивного шуму добре описує дію зернистості фотоплівки, флуктуаційний шум в радіотехнічних системах і т.д.

Якщо дія шуму позначається не по всій протяжності поля зображення, а тільки в випадково розташованих точках, в яких значення функції яскравості замінюються випадковими величинами, то шум називають імпульсним. На зображенні такі перешкоди виглядають ізольованими контрастними точками. Будемо вважати, що спотворені точки рівномірно розподілені по всьому полю зображення, а яскравість перекручених точок має рівномірний розподіл в деякому діапазоні. Імпульсний шум характерний для систем передачі зображень по радіоканалах з використанням нелінійних методів модуляції, а також для цифрових систем передачі і зберігання зображень. Зокрема, імпульсний шум притаманний пристроїв введення зображень з телевізійної камери.

2.1.2 Типи операцій

Існує величезна безліч різних операцій над растровими даними. Однак практично всі вони належать до одного з трьох типів операцій, наведених у таблиці 2.1. Така класифікація необхідна для того, щоб зрозуміти, якого результату можна очікувати від виконання тієї чи іншої операції і яких обчислювальних витрат це буде коштувати. При описі складності операції треба зауважити, що складність наводиться в розрахунку на один піксель зображення, розмір зображення дорівнює $N \times N$ пікселів, розмір області сусідніх пікселів - $R \times R$.

Таблиця 2.1 – Типи операцій над растровими даними

Тип операції	Опис	Складність на піксель
1	2	3
Точкова	Вихідне значення пікселя залежить тільки від вхідного значення пікселя.	const.

Продовження таблиці 2.1

1	2	3
Локальна	Вихідне значення пікселя залежить від значень сусідніх пікселів.	P^2
Глобальна	Вихідне значення пікселя залежить від значень пікселів всього зображення	N^2

Робота операцій трьох типів проілюстрована на рисунку 2.1. Тут кожна з операцій виконується над вихідним растровим зображенням $a[m, n]$ і переводить його в зображення $b[m, n]$. Розмірність результуючого зображення може бути іншою.

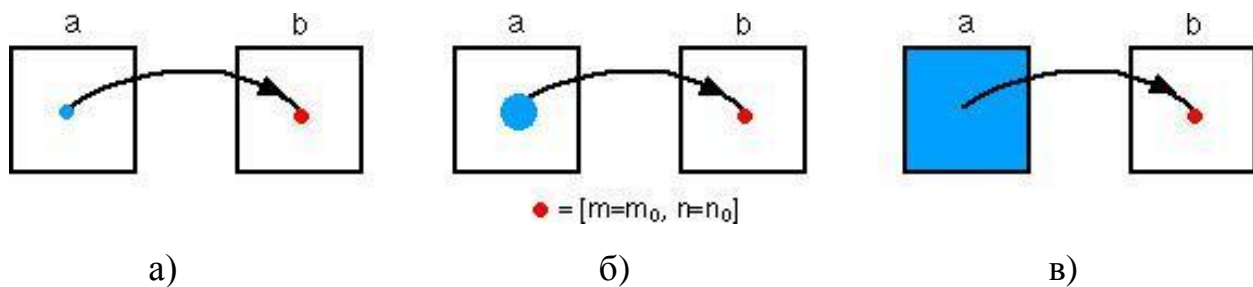


Рисунок 2.1 – Операції над растровими даними:

а – точкова; б – локальна; в – глобальна

В даному випадку піксель з координатами $[m_0, n_0]$ є поточним пікселем зображення, до якого застосовується перетворення.

При роботі з растрами великої розмірності тимчасові витрати на виконання локальних і глобальних операцій можуть бути досить великі, тому в сенсі швидкодії варто віддавати перевагу точковим операціями.

2.1.3 Зменшення кадрового розміру

Найпершим кроком на етапі попередньої обробки растрових даних часто є зменшення розміру кадру. Виконання даної операції на початковому етапі дає дві важливі переваги:

- істотне скорочення обсягу інформації, яку необхідно обробляти на кожному з наступних етапів.
- ослаблення адитивного та імпульсного шумів на вхідних даних.

Інакше цей крок можна назвати переходом до усереднених значень пікселів, оскільки значення компонентів кольору вихідного пікселя замінюються усередненими значеннями компонентів кольору даного пікселя і його сусідів. Результуючий піксель будемо називати мінімальної зоною (мінзоною) вихідного зображення, оскільки після цього перетворення вона стає мінімальною одиницею зображення, над якою будуть виконуватися всі подальші операції. Колір мінзони може обчислюватися, як середнє арифметичне значення кольорів 4, 16 і т.д. пікселів. Растр, отриманий, в результаті даного перетворення, будемо також називати картою мінзон. Для випадку усереднення чотирьох пікселів в одну мінзону перетворення запишеться так:

$$\begin{aligned}
 R_{ij} &= \frac{r_{2i,2j} + r_{2i,2j+1} + r_{2i+1,2j} + r_{2i+1,2j+1}}{4} \\
 G_{ij} &= \frac{g_{2i,2j} + g_{2i,2j+1} + g_{2i+1,2j} + g_{2i+1,2j+1}}{4}, \\
 B_{ij} &= \frac{b_{2i,2j} + b_{2i,2j+1} + b_{2i+1,2j} + b_{2i+1,2j+1}}{4}
 \end{aligned}
 \tag{2.1}$$

де R_{ij} , G_{ij} , B_{ij} - значення червоної, зеленої і синьої компонент кольору мінзони з i -го рядка і j -го стовпця, r_{ij} , g_{ij} , b_{ij} - значення червоної, зеленої і

синьої компоненти кольору пікселя з i -го рядка і j -го стовпця вихідного растра.

Здійснення переходу від чотирьох пікселів до однієї мінзони дозволяє в 4 рази скоротити обсяг інформації для обробки і в 16 разів – для випадку 16 пікселів, що може істотно збільшити швидкість подальшої обробки і знизити завантаження процесора.

Також необхідно зауважити, що це перетворення веде до помітного зниження рівня шуму. Причому, відбувається вплив як на імпульсний, так і на адитивний шум, оскільки усереднення дозволяє в рівній мірі згладжувати будь-які сплески значень кольору окремого пікселя за значеннями кольору сусідніх, незалежно від природи цих сплесків.

Однак важливо пам'ятати, що, незважаючи на всі перераховані переваги, виконання переходу до карти мінзон веде до втрати частини інформації про вихідний кадрі, тому усереднення по числу пікселів, більше 16, робити не рекомендується. Крім того, не варто застосовувати дане перетворення, якщо втрата дрібних деталей вихідного зображення неприпустима.

2.1.4 Лінійна фільтрація

Існує цілий клас методів ослаблення рівня зашумленості зображень, а також здатних виконувати інші операції (розмиття, виділення кордонів), заснованих на лінійній фільтрації вихідного зображення. Лінійна фільтрація передбачає застосування до пікселів растра просторово інваріантного перетворення, яке дозволило б підкреслити потрібні елементи зображення і уникнути впливу інших, менш значущих.

В основі лінійної фільтрації лежить операція згортки, яка для дискретного випадку записується, як у формулі (2.2):

$$c[m, n] = a[m, n] \otimes b[m, n], \quad (2.2)$$

де c – результуючий растр, отриманий після згортки, m , n – координати x і y , в яких виконується згортка, a , b – вихідні растри.

Береться ядро згортки, яке представляє собою матрицю малої розмірності (зазвичай не більше ніж 5×5 елементів), b – вихідне зображення, яке необхідно відфільтрувати.

Більш детально операція згортки розписана у формулі (2.3):

$$c[m, n] = \sum_{j=-m/2}^{u/2v} \sum_{k=-v/2} a[j, k] \times b[m - j, n - k], \quad (2.3)$$

де j , k – лічильники по горизонталі і вертикалі при обчисленні згортки, u , v – лінійні розміри ядра згортки (ширина, висота), інші позначення відповідають позначенням з формули (2.2).

Згортка обчислюється для кожного пікселя вихідного зображення. Залежно від того, яке ядро вибрано для згортки, результат може мати відчутні відмінності.

Існує безліч стандартних ядер згортки. Серед них можна виділити згортки для згладжування, розмиття, пом'якшення кордонів, підкреслення країв і т.д.

Варто зауважити, що сама операція обчислення згортки – операція, яка виконується за час порядку $O(n^2)$ – для найгіршого випадку. Однак найчастіше лінійні розміри растра помітно перевершують розміри ядра згортки, що робить фактичні трудовитрати істотно меншими, порівнянними з $O(n)$. Операція в цьому випадку відноситься до типу локальних.

При роботі в форматі RGB згортка виконується над кожною з трьох компонент кольору окремо, створюючи ефект обробки трьох окремих растрових зображень у відтінках сірого.

Одним з окремих випадків застосування лінійної фільтрації до відео зображень є здійснення згортки з метою згладжування. Згладжування вирівнює скачки яскравості на зображенні і дозволяє видалити зайвий шум.

Найчастіше для згладжування застосовуються ядра згортки розміром 3×3 , оскільки вони в більшості випадків дозволяють домогтися потрібного ефекту ціною порівняно високих, але все ж допустимих трудовитрат. Ядра згортки розмірності 5×5 , 7×7 і т.д. використовуються дуже рідко через надто великих трудовитрат. Наприклад, застосування згортки з ядром 5×5 , в порівнянні з випадком 3×3 дає в $(5/3) * (5/3) = 25/9 \approx 2.78$ разів більше операцій на обробку одного пікселя зображення.

Другий цікавий для нас спосіб застосування лінійної фільтрації до обробки відео зображень – пакунок з ядром для підкреслення країв (інакше Edge Detect). Такі ядра згортки дозволяють виділити контури об'єктів і придушити інші елементи зображення. Серед ядер згортки для цієї мети можна виділити кілька основних – це: Sobel Edge Detector, Gauss Edge Detector, Prewitt Edge Detector, Second-Derivatives filter.

Деякі ядра складаються з двох матриць, одна з яких виділяє вертикальні кордони, друга – горизонтальні. Інші ядра містять всього одну матрицю, структура якої дозволяє виділяти будь-які різкі переходи яскравості.

Великим недоліком методів виділення контурів, заснованих на лінійній фільтрації, є необхідність додаткової обробки відфільтрованого зображення. Особливо це справедливо для растрів з плавними переходами кольорів. Фільтрація практично ніколи не дає результуючого растра, з абсолютно чіткими кордонами і без зайвого «сміття». Для одного зображення фільтр може дати дуже товсті контури з великою кількістю зайвих елементів, для іншого контури будуть дуже тонкими, з великою кількістю розривів. Необхідність подальшої обробки може стати істотною проблемою, швидке і точне рішення якої не завжди можливо. Це стало однією з важливих причин, які змусили відмовитися від застосування даного методу для виявлення об'єктів.

2.1.5 Медіанна фільтрація

Медіанний фільтр, на відміну від фільтра, що згладжує, реалізує нелінійну процедуру придушення шумів. Медіанний фільтр являє собою ковзне по зображенню вікно W , що охоплює непарне число пікселів зображення [5]. Колір поточного пікселя замінюється медіаною кольорів всіх пікселів зображення, що потрапили у вікно. Медіаною дискретної послідовності x_1, x_2, \dots, x_L (для непарного L) називають такий її елемент, для якого існують $(L - 1) / 2$ елементів, менших або рівних йому за величиною, і $(L - 1) / 2$ елементів, великих або рівних йому за величиною. Іншими словами, медіаною є середній по порядку член ряду, що виходить при впорядкуванні вихідної послідовності.

Як і згладжувальний фільтр, медіанний фільтр використовується для придушення адитивного та імпульсного шуму на зображенні. Характерною особливістю медіанного фільтра, що відрізняє його від згладжувального, є збереження перепадів яскравості (контурів). Особливо ефективним медіанний фільтр є в разі імпульсного шуму. На рисунку 2.2 показано вплив згладжувального і медіанного фільтрів з трьохелементним вікном на зашумленій адитивним шумом перепадом яскравості уздовж однієї координати растрового зображення.

Що стосується імпульсного шуму, то, наприклад, медіанний фільтр з вікном 3×3 повністю пригнічує поодинокі спалахи на рівномірному фоні, а також групи з двох, трьох і чотирьох імпульсних викидів. У загальному випадку для придушення групи імпульсних перешкод розміри вікна повинні бути, щонайменше, вдвічі більше розмірів групи перешкод.

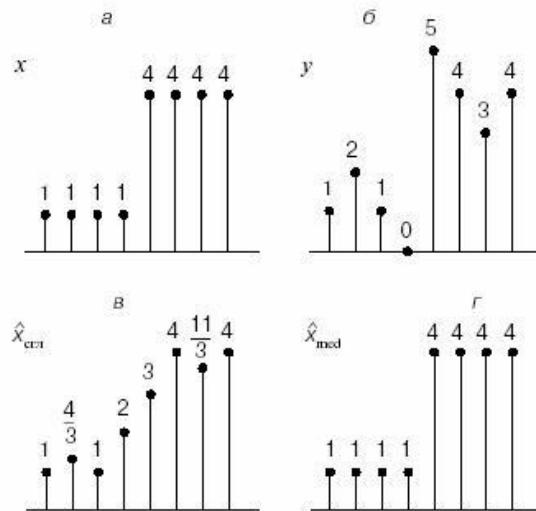


Рисунок 2.2 – Приклад роботи згладжувального і медіанного фільтрів: а – вихідний сигнал; б – зашумлений сигнал; в – результат роботи фільтра, що згладжує; г – результат роботи медіанного фільтра

Поряд з важливою перевагою медіанного фільтра, що полягає в тому, що він істотно менше змащує контури об'єктів на зображенні, даний підхід має і суттєвий недолік, пов'язаний з тим, що обчислення медіани вимагає високих обчислювальних витрат. Адже, незважаючи на те, що трудомісткість обчислення медіанного елемента масиву в середньому – є близько $O(n)$, доводиться вдаватися до додаткових хитрувань, щоб на практиці дійсно домогтися лінійної складності. Але для найгіршого випадку складність все одно досягає $O(n^2)$, особливо якщо число елементів в масиві невелика.

2.1.6 Помірна фільтрація

Метод помірної (консервативної) фільтрації (conservative smoothing) багато в чому повторює ідею медіанного фільтра, однак відмінність в роботі і ефекті від обробки цими фільтрами є [6]. Помірний фільтр в процесі обробки також використовує ковзне по зображенню вікно. Найбільш часто використовується вікно розміром 3×3 пікселя, всередині якого обчислюється максимальне і мінімальне значення інтенсивності кольору (рисунок 2.3).

Іншими словами, якщо значення інтенсивності компоненти кольору оброблюваного пікселя потрапляє між максимальним і мінімальним значеннями, взятими по пікселям вікна, то інтенсивність залишається без змін, в іншому випадку, якщо це значення менше мінімального, то воно стає рівним мінімальному, якщо більше максимального – максимальному. Таке перетворення дозволяє відсікти неприпустимо великі і малі значення інтенсивності, які можуть виникати внаслідок дії компоненти шуму. Оскільки пікселі випадкового шуму рідко утворюють великі групи, то фільтрація даними алгоритмом справляється з ним без особливих зусиль.

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Max: 127
Min: 115

Рисунок 2.3 – Вікно помірною фільтра, максимальне і мінімальне значення серед елементів вікна

Зауважимо, що алгоритм не змінює пікселів, значення інтенсивностей яких поєднуються зі значеннями сусідніх пікселів. Це означає, що зображення, на якому шум відсутній, буде дуже слабо спотворюватися такою обробкою. Крім того, будуть зберігатися лінії контурів. Дана операція відноситься до типу локальних, тому трудомісткість її виконання складе порядок $O(n)$.

2.1.7 Міжкадрова фільтрація

Ще один спосіб зниження рівня зашумленості відеокадрів – це міжкадрова фільтрація. Таку назву метод отримав з тієї причини, що його робота заснована на аналізі змін між кадрами відеоряду і фільтрації тих з цих змін, які, ймовірно, викликані зашумленням растрових даних.

Це простий, нетрудоїмкий, але ефективний метод згладжування адитивного шуму. Алгоритм заснований на тому припущенні, що значення кольору відповідних пікселів на сусідніх кадрах не можуть сильно відрізнятися, якщо в цій області відсутній рух. Якщо значення кольору різняться несуттєво, значить ця різниця викликана адитивним шумом при реєстрації зображення, і значення кольору даного пікселя на поточному кадрі замінюється усередненим значенням даного пікселя і відповідного пікселя на попередньому кадрі. В іншому випадку дуже ймовірно, що дана відмінність викликана наявністю в цій області рухомого об'єкту, і значення даного пікселя залишається без зміни. Величина максимальної відмінності компоненти кольору відповідного пікселя на поточному та попередньому кадрах задається порогом порівняння. Лінійні розміри кадру залишаються незмінними.

Перевагою алгоритму є його швидкодія. Він працює за один прохід, не вимагаючи додаткової обробки і накопичення інформації.

2.1.8 Коригування яскравості

Ми розглянули деякі способи видалення шуму з вихідного зображення. Однак етап попередньої обробки відеокадрів на цьому зазвичай не закінчується. Видалення шуму дозволяє мінімізувати кількість помилкових спрацьовувань детектора руху на перешкоди, що виникають при реєстрації та передачі зображень, але кадри, що містять одну і ту ж сцену, після цього все ще можуть помітно відрізнятися. Причиною цієї відмінності є зміна рівня

освітленості при реєстрації різних кадрів. Зміна рівня освітленості може бути викликана включенням або виключенням штучного освітлення або зміною погодних умов, якщо зйомка проводиться поза приміщенням. Зазвичай в цей момент поелементні різниці двох сусідніх кадрів досягають дуже великих значень, що веде до помилкового спрацьовування детектора, що фіксує рух по всьому простору зміни освітлення. Щоб уникнути подібних помилкових спрацьовувань, на етапі попередньої обробки доводиться вдаватися до коригування рівнів яскравості відеокадрів [7, 8].

Якби зміна рівня освітленості була відома заздалегідь і могло бути задана функціонально, на етапі попередньої обробки можна було б ліквідувати її шляхом зворотного перетворення. Однак спрогнозувати подібні зміни заздалегідь майже неможливо, тому для їх виявлення необхідно аналізувати самі растрові дані в припущенні, що всі зміни між сусідніми кадрами були викликані саме зміною рівня освітленості, а не зміною сцени.

В обчисленні яскравостей пікселів поточного кадру будемо спиратися на пікселі попереднього кадру:

$$I_2(x, y) = a(x, y) \cdot I_1(x, y) + b(x, y), \quad (2.4)$$

де $I_1(x, y)$ – значення яскравості пікселя попереднього кадру, $I_2(x, y)$ – значення яскравості пікселя поточного кадру, $a(x, y)$, $b(x, y)$ – коефіцієнти лінійного перетворення.

У загальному випадку a і b є масивами даних, що містять різні значення для кожного з пікселів. На практиці ця вимога зазвичай спрощується і замінюється алгоритмом обробки вікном площею кілька десятків пікселів, всередині якого обчислюються значення коефіцієнтів a і b , після чого проводиться коригування цими значеннями пікселів, що потрапляють у вікно на поточному кадрі. Потім вікно зсувається в нове положення, таким чином, обходячи весь кадр.

Для випадку рівномірної зміни освітленості вздовж усього растра можна вважати, що рівні яскравості всіх пікселів змінюються рівномірно, а вектори коефіцієнтів a і b замінити просто скалярними величинами.

2.1.9 Коригування контрастності

Важливим кроком при підготовці відеокадрів до виявлення на них рухомих об'єктів є коригування рівня контрастності зображень. Таке коригування полягає в аналізі і перетворенні гістограми яскравостей пікселів вихідного зображення таким чином, щоб її форма ставала більш рівномірною і зникав зсув щодо середнього значення інтенсивності. Рівномірність і незрушність гістограми інтенсивностей пікселів зображення необхідні для поліпшення роботи алгоритмів виявлення, заснованих на обчисленні попідсельно різниць між кадрами.

Зрушність гістограми ліквідується шляхом лінійного контрастування, ідея якого полягає в перетворенні інтенсивності кожного з пікселів зображення щодо максимального і мінімального значень в гістограмі для поточного кадру.



Рисунок 2.4 – Вихідна гістограма

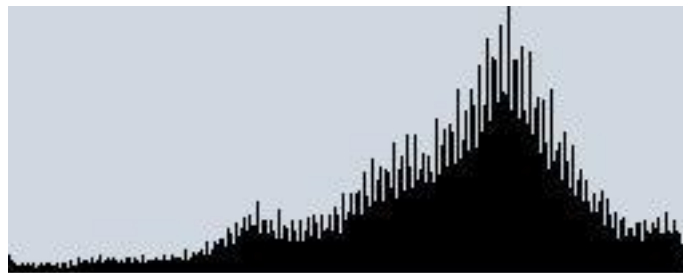


Рисунок 2.5 – Гістограма після лінійного контрастування

Потрібно зауважити, що такий підхід буде дуже рідко приводити до оптимального результату, оскільки наявність найменшого імпульсного шуму дасть ефект появи на зображенні пікселів з різними значеннями інтенсивностей, яких не було на оригінальному документі, і ефект від лінійного контрастування істотно зменшиться.

Крім збільшення контрасту, на етапі підготовки зображень відеокадрів до обробки різницевим детектором руху, дуже важливо привести ці зображення до одного виду. Будемо розуміти під цим таке перетворення вихідного зображення, після якого гістограма інтенсивностей кольорів стає рівномірною для всіх значень інтенсивності, що означає, що всі значення стають рівноімовірними [6].

Описаний ефект досягається шляхом заміни інтенсивності кольору пікселя значенням функції розподілу за однакової кількості аргументу застосовується для розрахунку інтенсивності з коефіцієнтом масштабування на наявне в розпорядженні кількість рівнів інтенсивності:

$$\begin{aligned} k &= u[m, n] \\ u'[m, n] &= (b - a)n_k / N^2, \end{aligned} \quad (2.5)$$

де $u[m, n]$ – піксель вихідного зображення з координатами (m, n) , $u'[m, n]$ – піксель результуючого зображення з координатами (m, n) , a – мінімально можливе значення інтенсивності, b – максимально можливе значення інтенсивності, n_k – кількість пікселів в оригінальному документі з рівнем інтенсивності, що дорівнює або менше k , N – загальна кількість пікселів зображення.

Після обробки за допомогою даного методу всі кадри відеоряду стануть схожими за рівнем освітленості (причому розподіл кількості пікселів по значенням інтенсивності стане близьким до рівномірного), що запобіжить помилковому спрацьовуванню різнісного детектора руху.

2.2 Виявлення руху

Задача виявлення рухомих об'єктів була поставлена досить давно, але в силу своєї специфіки вона до цих пір не має однозначного вирішення.

По-перше, умови виявлення можуть бути різні. Наприклад, якщо мати справу з бінарним чорно-білим зображенням, то завдання істотно спрощується в порівнянні з випадком, коли на вхід надходять повнокольорові растри з плавними переходами кольору від одного відтінку до іншого. Зображення може надходити з статично розташованої камери, таким чином, всі кадри будуть мати приблизно однаковий фон з можливими відзнаками рівня освітлення. З іншого боку, камера, розташована на об'єкті, що рухається, може знімати інший, нерухомий об'єкт. Рівень шуму може істотно варіюватися. Природні явища, такі, як дощ, сніг, туман, вітер і т.д. можуть вносити помітний елемент коливань в спочатку нерухому сцену. Все це робить алгоритми, які відмінно працюють в одних умовах, абсолютно непридатними для інших.

Від алгоритмів, що працюють в автоматизованих системах виявлення руху, – особливо це стосується систем контролю периметра, інших систем виявлення вторгнення – зазвичай потрібна стійкість роботи в широкому діапазоні істотно різних зовнішніх умов. У загальному випадку вимоги до подібних алгоритмів виглядають наступним чином:

Низька обчислювальна складність і робота в реальному часі. Це, мабуть, найважливіша характеристика, яка оцінюється в першу чергу при виборі детектора руху. Її виняткова важливість обґрунтована тим, що практично будь-яка система охорони має в своєму розпорядженні більше однієї камери, дані якої необхідно обробляти. Це означає, що на наявній апаратурі буде працювати далеко не один детектор. Якщо ж обробка даних від однієї камери буде займати занадто багато часу, то навіть найменша система, що обробляє дані від 2-4 камер, стане розподіленою, оскільки потужності однієї ЕОМ для обробки всіх даних не вистачить.

Обслуговування розподіленої системи буде обходитися власникові істотно дорожче, не рахуючи того, що вартість апаратури і програмного комплексу збільшиться в кілька разів.

- стійка робота в будь-який час року при будь-яких погодних умовах.
- мінімальна кількість помилкових спрацьовувань на шум.
- мінімальна кількість спрацьовувань на об'єкти, що не представляють небезпеки для об'єкту, що охороняється (хитні дерева, листя, інша рослинність та ін.).

Вимоги до систем виявлення руху, крім перерахованих вище, що стосуються алгоритмів виявлення руху, зазвичай виглядають наступним чином:

- Багатоканальність. Цей пункт кілька дублює вимога до алгоритмів виявлення руху про низьку обчислювальну складність, однак вкрай важливий і для системи в цілому.

- Масштабованість. Система повинна давати можливість легко збільшувати кількість джерел вхідних даних.

- Можливість гарячої заміни. Необхідно мати можливість підключення / відключення будь-якої камери на льоту, без зупинки роботи інших каналів.

- Відсутність жорсткої прив'язки до певного типу джерел вхідних даних. Бажано, щоб система могла працювати з різними типами камер, починаючи від чорно-білих, закінчуючи кольоровими.

2.3 Міжкадрова різниця

Обчислення міжкадрової різниці є дуже поширеним методом первинного виявлення руху, після виконання якого, власне кажучи, вже можна сказати, чи присутній в потоці кадрів рух. До недавнього часу багато детектори руху функціонували саме за таким принципом [10]. Однак такий підхід дає досить грубу оцінку, приводячи до наявності неминучою

помилкової реакції детектора на шум записуючого устаткування, зміну умов освітлення, легке хитання камери тощо. Таким чином, відеокадри повинні бути попередньо оброблені перед обчисленням різниці між ними. Відомі способи попередньої обробки відеокадрів ми розглянули раніше, тому тепер займемося власне обчисленням міжкадрової різниці.

Алгоритм обчислення міжкадрової різниці двох кадрів для випадку обробки кольорового відео в форматі RGB виглядає наступним чином:

1) На вхід алгоритму надходять два відео кадру, що представляють собою дві послідовності байт в форматі RGB.

2) Проводиться обчислення попіксельно міжкадрових різниць за наступною схемою:

$$\begin{aligned} R_{res}^i &= |R_1^i - R_2^i| \\ G_{res}^i &= |G_1^i - G_2^i|, \\ B_{res}^i &= |B_1^i - B_2^i| \end{aligned} \quad (2.6)$$

де R_{res}^i , G_{res}^i , B_{res}^i – значення червоної, зеленої і синьої компоненти кольору i -го пікселя результуючого растра, R_1^i , G_1^i , B_1^i , R_2^i , G_2^i , B_2^i – значення червоної, зеленої і синьої компоненти кольору i -го пікселя на першому і другому кадрі.

3) Для кожного пікселя обчислюється середнє значення між значеннями трьох компонент кольору:

$$p^i = (R_{res}^i + G_{res}^i + B_{res}^i) / 3. \quad (2.7)$$

4) Середнє значення порівнюється із заданим порогом. В результаті порівняння формується двійкова маска:

$$m^i = \begin{cases} 0, p^i \leq T \\ 1, p^i \geq T \end{cases}, \quad (2.8)$$

де m^i – значення i -го елемента маски, T – поріг порівняння, іноді званий також порогом або рівнем чутливості.

Таким чином, на виході алгоритму формується двійкова маска, одному елементу якої відповідають три компоненти кольору відповідного пікселя вихідних двох кадрів. Одиниці в масці розташовуються в областях, де, можливо, присутній рух, однак на даному етапі можуть бути і помилкові спрацьовування окремих елементів маски.

В якості двох вхідних кадрів можуть використовуватися два послідовних кадру з потоку, однак можливе використання кадрів з великим інтервалом, наприклад, рівним 1-3 кадру. Чим більше такий інтервал, тим вище чутливість детектора до малорухливим об'єктів, які відчують лише вкрай малий зсув за один кадр і можуть відсікатися, будучи віднесеними до шумової складової зображення.

Перевагою даного методу є простота і невимогливість до обчислювальних ресурсів. Метод широко застосовувався раніше через те, що в розпорядженні розробників не було достатніх обчислювальних потужностей. Однак і зараз він широко використовується, особливо в багатоканальних охоронних системах, коли необхідно обробляти сигнал від декількох камер на одній ЕОМ. Адже трудомісткість алгоритму має порядок $O(n)$ і здійснюється всього за один прохід, що дуже важливо для растрів великої розмірності, таких як 640x480 пікселів, 768x576 точок, з якими нерідко працюють сучасні відеокамери.

2.3.1 Метод базового кадру

Описуючи метод міжкадрової попінкельної різниці з побудовою маски руху, в якості двох вхідних кадрів ми ставили або два сусідніх кадру, або два

кадри, взятих з невеликим інтервалом. Однак використання цього методу залишає можливість обчислювати різницю з деяким кадром, який би містив виключно нерухомі області фону (базовий кадр). Такий підхід дав би нам істотне збільшення ймовірності виявити будь-який об'єкт, як самий повільний, так і швидкий, причому саме в тій точці, в якій він знаходився в даний момент. Інакше даний метод називається методом віднімання або сегментації фону [11].

Робота методу повністю аналогічна роботі алгоритму міжкадрової різниці з тією лише різницею, що різниця обчислюється між поточним і базовим кадром. Великою проблемою тут є спосіб побудови базового кадру, оскільки він повинен володіти декількома властивостями:

Якщо кадр являє собою кадр реального зображення, він повинен мінімально відстояти за часом від поточного кадру. Зауважимо, що в реальних умовах це не завжди можливо з тієї причини, що зйомка може вестися в місцях частого появи в кадрі рухомих об'єктів, тому система буде дуже рідко фіксувати сцену з повною відсутністю в ній руху, внаслідок чого вона не матиме можливості оновлювати базовий кадр досить часто.

– Якщо базовий кадр готується штучно, він повинен містити мінімальну кількість рухомих елементів, інакше неминучі помилкові спрацьовування на об'єкти, яких на поточному кадрі вже немає, проте базовий кадр містить якісь їхні елементи.

– Мінімальний рівень шуму. Перед оновленням базового кадру необхідно проводити фільтрацію.

Існує два підходи до побудови базового кадру.

Перший заснований на кумулятивному його накопиченні з використанням всіх кадрів потоку. При такому способі побудови базовий кадр неминуче містить елементи, що рухаються, однак при обчисленні різниці між поточним і базовим кадром відмінності в інтенсивності відповідних пікселів, що належать рухається об'єкту, все ж є значним і дозволяють виявити рух. Плюсом такого способу є його простота. Однак при

обчисленні різниці необхідно виконати додаткові методи шумозаглушення [12], оскільки поріг порівняння при побудові маски руху в даному випадку не може бути високим (в силу специфіки базового кадру). В іншому випадку рух буде просто втрачено, а при невеликих значеннях порога маска руху буде неминуче фіксувати велику кількість пікселів шуму.

Другий підхід виглядає більш інтелектуальним, оскільки при його використанні базовий кадр будується виключно з нерухомих областей, взятих з кожного з поточних кадрів. Для цього на вході алгоритму необхідно мати вихідні дані детектора рухомих об'єктів від обробки попереднього кадру. Використовуючи цю інформацію, алгоритм позначає області, де на попередньому кадрі були виявлені рухомі об'єкти, після чого при попіксельно копіюванні поточного кадру в базовий пікселі помічених областей пропускаються. Таким чином, базовий кадр ніколи не містить ніяких елементів рухомих об'єктів, що досягається ціною порівняно невеликого ускладнення алгоритму.

2.3.2 Математична морфологія

Математична морфологія представляє собою аналіз зображення з точки зору його форми. Робота алгоритмів, заснованих на застосуванні даного підходу, пов'язана з виконанням над зображенням ряду перетворень, що змінюють форму об'єктів, з-тримаються на зображенні [13].

Математична морфологія застосовується в різних системах, що мають справу з обробкою зображень, на різних етапах і для досягнення різних цілей:

- покращення візуальних характеристик зображення (яскравість, контрастність та ін.);
- відновлення зіпсованих зображень, наприклад, реставрація фотознімків;
- виявлення контурів;

– зниження рівня шуму.

Операції математичної морфології можуть проводитися над кольоровими, чорно-білими зображеннями і зображеннями в відтінках сірого. Для цих трьох випадків формально вони визначаються дещо по-різному. Ми розглянемо випадок бінарних чорно-білих растрів. Такий вибір зроблено з тієї причини, що застосування нами операцій математичної морфології на даному етапі має на меті додаткової фільтрації шуму в масці руху. Оскільки маска руху являє собою растр, що складається з нулів і одиниць, нам достатньо обмежитися розглядом саме випадку роботи з бінарними зображеннями.

В основі базових операцій математичної морфології для бінарних зображень лежать операції з теорії множин. Для початку розглянемо дві операції: розширення (dilation) і ерозію (erosion). Формально вони можуть бути визначені в такий спосіб:

$$\begin{aligned} D(A, B) &= A \oplus B = \bigcup_{\beta=B} (A + \beta), \\ E(A, B) &= A \otimes B = I(A \cdot \beta) \end{aligned} \quad (2.9)$$

де A – растр вихідного зображення, B – структурований елемент, β – піксель структурованого елемента, $D(A, B)$ – результуючий растр після впливу операцією розширення, $E(A, B)$ – результуючий растр після впливу операцією ерозії. Дія операції розширення полягає в збільшенні площі об'єкта, ерозії – навпаки, в її зменшенні. Величина і спосіб зміни площі об'єкта залежить від вибору структурного елемента.

2.3.3 Виділення об'єктів

Ми підійшли до того, щоб визначити поняття об'єкта з точки зору детектора руху. Воно буде дещо відрізнятися від того, що людина звикла розуміти під об'єктом, сприймаючи оком навколишні його предмети.

Виділяючи об'єкти на зображенні, людське око працює в зв'язці з головним мозком, який на поточний момент вже зберігає достатньо інформації про типи, форми, лінійні розміри, кольорі та інших характеристиках, властивих тим чи іншим об'єктам, з якими людина стикалася раніше. Таким чином, поряд з виділенням об'єктів сцени, людський мозок виконує процедуру їх розпізнавання, тому людина без праці розрізняє відомі йому об'єкти.

Робота по виділенню об'єктів детектором руху, заснованим на алгоритмі обчислення попиксельно різниць між кадрами, починається з аналізу маски руху. Реальним об'єктам в ній зазвичай відповідають пікселі (або мінзони), які утворюють зв'язкові групи, тому логічно визначити об'єкт з точки зору детектора руху як групу зв'язкових пікселів в масці руху. Такому об'єкту відповідають декілька параметрів:

- 1) Лінійні розміри мінімального прямокутника (надалі просто прямокутник), який можна описати близько пікселів групи.
- 2) Координати (x, y) центральної точки прямокутника. Будемо вважати дану точку центральною точкою об'єкта.
- 3) Кількість пікселів, що входять в групу.
- 4) Область маски руху, що лежить усередині прямокутника.
- 5) Область поточного кадру, що лежить усередині прямокутника.
- 6) Оскільки детектор виявляє тільки рухомі об'єкти, ця група пікселів і відповідний їй об'єкт з надходженням на вхід детектора нових кадрів будуть відчувати зсув, тому крім параметрів, перерахованих вище, можна ввести ще кілька:
 - 7) Вектор, що описує напрям і швидкість руху об'єкта.
 - 8) Масив, що містить координати центру об'єкта на попередніх кадрах.
 - 9) Час життя об'єкта, що вимірюється в кількості кадрів.

Виділення об'єктів відбувається при послідовному обході пікселів маски руху. Якщо в масці руху виявлена одиниця, запускається процедура пошуку пікселів, зв'язкових з даними, яким також відповідають одиниці в

масці руху. Знайдені пікселі позначаються, щоб уникнути повторного їх виявлення. Алгоритм пошуку може бути рекурсивним або хвильовим.

2.3.4 Рекурсивний пошук

Алгоритм рекурсивного пошуку являє собою просту процедуру, на вхід якої надходить поточний піксель. У середині процедури для всіх сусідів даного пікселя здійснюється перевірка, чи відповідає кожному з них одиниця в масці руху і чи не є він уже належить будь-якого об'єкта. Якщо сусідній піксель проходить перевірку, то він стає поточним і для нього рекурсивно запускається процедура пошуку. Сусіди пікселя можуть обиратися за принципом 4-зв'язності або 8-зв'язності. У першому випадку перевіряється 4 сусіда, розташованих вище, нижче, ліворуч і праворуч від поточного пікселя. У другому випадку перевірка зачіпає всі 8 пікселів маски оточуючих поточний піксель.

Приклад обходу зв'язкової групи пікселів можна проілюструвати рис. 2.6(a). Тут використовується алгоритм з перевіркою 8 сусідніх пікселів. Порядок перевірки наведено на рис. 2.6(б).

Недоліком рекурсивного пошуку є можливе переповнення стека при рекурсивному виклику процедури пошуку. Імовірність переповнення тут велика, оскільки максимально можлива кількість рекурсивних викликів (в разі, якщо одиниці будуть заповнювати всю маску руху) може досягати величини, близької кількості пікселів. Для випадків обробки зображень з високою роздільною здатністю, наприклад, 640x480 пікселів, процедура пошуку може рекурсивно викликатися близько 300 тисяч разів. Зазвичай ця величина на кілька порядків менше, проте щоб гарантовано уникнути переповнення стека, рекомендується обмежувати можливий розмір об'єкта, а значить і глибину рекурсії, зверху або використовувати хвильовий алгоритм пошуку.

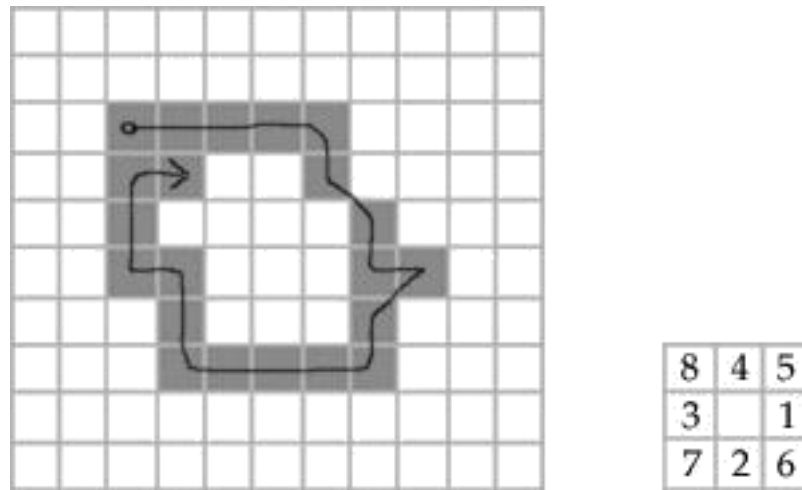


Рисунок 2.6 – Рекурсивний пошук: а) послідовний прохід; б) перевірка сусідніх пікселів

2.3.5 Пошук хвильовим методом

Хвильовий метод отримав таку назву з тієї причини, що кожна з ітерацій його роботи нагадує хвилю, що піднялася на поверхні води від впавшого у неї каменя. Хвилі розходяться від точки падіння каменя радіально у вигляді концентричних кіл. Аналогічним чином відбувається обхід пікселів маски і в алгоритмі хвильового пошуку. З точки, де розташований поточний піксель (перший знайдений піксель, якому відповідав би одиниця в масці руху), запускається хвиля в усіх напрямках, число яких залежить від обраного типу зв'язності (4-зв'язний, 8-зв'язний). Для всіх пікселів, які досягаються хвилею в поточній ітерації, проводиться перевірка умов наявності одиниці в масці руху і відсутність позначки про входження в інший об'єкт. Всі пікселі, які успішно пройшли перевірку, заносяться в масив. На наступній ітерації хвиля запускається з усіх пікселів, що містяться в масиві. Масив очищається, і в нього заносяться пікселі, знову досягнуті хвилею. Всі оброблені пікселі позначаються як увійшли в поточну групу. При реалізації даного алгоритму зручно мати масив змінної довжини для приміщення в нього пікселів хвилі на поточній ітерації, або заздалегідь обчислювати максимально допустимий розмір цього масиву.

2.3.6 Трасування

Термін трасування має на увазі відстеження об'єктів, які рухаються і значень його параметрів протягом усього часу його перебування в кадрі. При правильному трасування об'єкта з'являється можливість перегляду траєкторії руху об'єкта з будь-яким рівнем деталізації (з точністю до кадру, до 3, 5 і т.д. кадрів). Така можливість забезпечується збереженням положення центральної точки об'єкта на кожному з кадрів в спеціальному масиві, який ми ввели для цієї мети при визначенні нами параметрів для кожного з рухомих об'єктів. Алгоритми, трасуючі рухомі об'єкти, не повинні плутати один об'єкт з іншим, втрачати об'єкт, якщо він зупинився на невеликий проміжок часу або зник за якою-небудь перешкодою (наприклад, людина, що сховався за деревом), а потім з'явився знову. Для цих цілей чудово підходить застосування нормалізованої функції взаємної кореляції, описаної в попередньому параграфі. Вона допомагає порівняти об'єкт з одного з попередніх кадрів з об'єктом, виявленим на знову отриманому кадрі і зробити висновок про їх схожості. Навіть якщо об'єкт, який можна було порахувати зниклим після того, як він пропав за якою-небудь перешкодою, перестав виявлятися, застосовуючи алгоритм обчислення функції кореляції можна виявити його знову на одному з наступних кадрів і продовжити трасування, взявши параметри об'єкта в момент його останнього виявлення і скорегувавши їх за поточним кадром. Це дуже важливо для побудови нерозривного траєкторії руху.

При обчисленні функції кореляції з метою виявлення на поточному кадрі об'єкта, виявленого раніше, використовуються тільки ті пікселі зображення, яким в масці руху відповідають одиниці. В іншому випадку в формулу для обчислення функції кореляції можуть потрапити значення кольорів пікселів, що лежать всередині прямокутної області, але за межами об'єкта, а ці пікселі зовсім не обов'язково повинні корелювати між собою, отже, значення функції кореляції в цьому випадку зменшиться. А якщо воно

стане менше обраного порога схожості, алгоритм може помилково порахувати дану прямокутну область растру несхожою на об'єкт, з яким проводиться порівняння, навіть якщо насправді це не так.

Очевидно, що обчислювати функцію кореляції для двох об'єктів не має сенсу, якщо лінійні розміри відповідних їм прямокутників істотно розрізняються, оскільки і так очевидно, що це різні об'єкти. Також безглуздо, наприклад, шукати в правому нижньому кутку поточного кадру об'єкт виявлений в лівому верхньому кутку попереднього кадру, оскільки ніякої цікавий для нас об'єкт не може за такий короткий час так істотно змінити своє місце розташування. Такі припущення можуть істотно скоротити витрати на трасування, в загальному випадку включають обчислення функції кореляція кожного об'єкта з кожним – $O(n^2)$. Крім того, для об'єкта, що зазнав вкрай мале зміщення на сусідніх кадрах можна взагалі не обчислювати значення функції кореляції. Адже ще до виконання даної операції стає очевидним, що два відбитки в одній і тій же області маски руху на сусідніх кадрах належать одному і тому ж об'єкту.

2.3.7 Фільтрація об'єктів за їх параметрами

Після того, як фільтр детектора руху вже здатний виявляти і відстежувати рухомі об'єкти, накопичуючи інформацію про них, реєструючи їх параметри, можна перейти до останнього етапу виявлення: відсікання не цікавить користувача об'єктів по їх параметрами.

Найпростіший тип можливої фільтрації об'єктів - по їх лінійними розмірами. Даний тип фільтрації може використовуватися для відсікання шуму, який все ж може залишатися на зображенні після всіх кроків етапу предоброботки і навіть після проведення фільтрації операціями математичної морфології. Для здійснення фільтрації даного типу необхідно задати мінімальні і / або максимальні розміри об'єктів, що цікавлять користувача. Після чого всі об'єкти, розміри яких будуть виходити за допустимі рамки,

будуть відсікатися, не даючи помилкової тривоги. Крім видалення залишків шуму, це може бути корисно для виключення спрацьовування системи охорони на птахів, тварин (при завданні мінімально допустимих розмірів) або на автомобілі (при завданні максимально допустимих розмірів).

Крім розмірів, допустимими рамками може обмежуватися швидкість і напрямок руху об'єкта. Це може бути корисно при спостереженні за ділянкою території, на якій можлива поява людей, що рухаються пішки, однак заборонений біг або швидка їзда на автомобілі, або заборонено рух в певному напрямку. Для виявлення випадків тривоги тут необхідно лише порівняти швидкість і напрямок руху об'єкта з забороненим напрямком і максимально допустимою швидкістю.

Крім того, ситуацією, при якій виникає тривога, може бути просте об'єднання кількох об'єкта в певній області кадру, яка є охороняється і поява будь-яких об'єктів там заборонено. Завдання такої області може здійснюватися за допомогою введення багатокутників, які окреслюють заборонену зону. Це випадок відсікання об'єктів по координаті їх центральної точки в кадрі.

Перераховані методи об'єктної фільтрації найчастіше необхідні для зручності користування системою виявлення рухомих об'єктів, оскільки іноді, наприклад, буває просто неможливо розмістити камеру стеження таким чином, щоб вона захоплювала тільки периметр, де знаходження рухомих об'єктів заборонено, і в кадр так чи інакше потрапляють об'єкти, рухомі за межами цього периметра, де саме по собі знаходження рухомих об'єктів не є забороненим. Однак наявність цих об'єктів буде приводити до постійного виникнення помилкової тривоги, що негативно позначиться на пильності оператора.

2.4 Розробка алгоритму фіксації руху

Основна мета даної роботи – розробка детектора рухомих об'єктів, відповідного основним вимогам, що пред'являються до алгоритмів, що працюють у складі охоронних систем, а саме, що забезпечує стійку роботу в широкому діапазоні навколишніх умов, мінімальна кількість помилкових спрацьовувань і високу швидкодію. Детектор повинен приймати на вхід кадри відеоряду у вигляді растрових зображень у форматі RGB, отримані з кольоровою або чорно-білою відеокамери або завантажуються з відеофайлу, обробляти ці дані, виявляючи на них рухомі об'єкти, виробляти відображення виділених об'єктів, траєкторій їх руху на відеокадрах і подавати їх на вихід.

У попередніх розділах нами були розглянуті основні методи попередньої обробки відеозображень і способи виявлення рухомих об'єктів різницевим детектором по бінарної масці руху. Велика частина цих методів давно відома, а їх придатність для обробки різних відеозображень досить добре досліджена. Однак специфіка роботи полягає в тому, що дана проблемна область є вкрай популярною сьогодні, а застосовність алгоритмів для неї, хоча і, ймовірно, широко досліджена, але не завжди широко висвітлена в літературі. Крім того, найважливіше питання – питання швидкодії при описі методів в статтях часто залишається в тіні, або про це говориться дуже мало.

Для якісної оцінки швидкості роботи алгоритму тут не завжди буває достатньо сказати про порядок його складності. Нерідкі випадки, коли швидкості роботи двох алгоритмів, обидва з яких мають лінійну трудомісткість, відрізняються в 2-4 і більше разів. Дуже часто така різниця не принципова і, часом, може бути навіть не помічена. В даному ж випадку прискорення роботи алгоритму в 2-4 рази означає, що за один і той же час на тій же ЕОМ може оброблятися 2-4 каналу відео замість одного. Не дивно, що правильний вибір алгоритму тут дуже важливий, навіть якщо це дасть приріст всього 10-15% до швидкості його роботи.

2.4.1 Предобробка

На етапі попередньої обробки було необхідно підготувати відеозображення до виявлення на них рухомих об'єктів. Можна вважати, що попередня обробка проведена якісно, якщо будь-які два відеокадра, отримані з невеликим інтервалом від однієї і тієї ж камери в умовах відсутності руху в зоні видимості камери є максимально схожими один на одного. Схожість кадрів можна визначити, обчисливши значення функції їх взаємної кореляції або побудувавши маску руху після обчислення міжкадрової різниці. Крім того, важливо мінімізувати витрати часу на проведення такої обробки.

В якості першого кроку було вирішено використовувати перехід від пікселів зображення до мінзонам, які приймають усереднені значення інтенсивностей кольорів, які входять до їх пікселів. Зрозуміло, що при такому переході втрачається досить велика частина інформації про оригінальний документ, тому, щоб уникнути втрат, які були б згубними для растрів малої розмірності (менших, ніж 640x480 пікселів), таке перетворення над ними не виконується. Для растрів розміром 640x480 пікселів і більше виконується перехід до мінзонам, який заміняє області 2x2 пікселя.

У підсумку, на цьому кроці маленькі зображення не відчують ніякого перетворення, а площа великих скорочується в 4 рази. Це означає, що на всіх наступних етапах буде оброблятися в 4 рази менше даних, що істотно заощадить процесорний час. Поряд з цим у великих зображеннях відбувається деяке зменшення рівня шуму, тому що вплив окремих пікселів випадкового шуму нівелюється їх сусідами.

Другим кроком на етапі попередньої обробки виконується міжкадрового фільтрація. Застосовується алгоритм, описаний в другому розділі. Його важлива перевага над іншими способами фільтрації полягає в тому, що метод має на увазі застосування точкових операцій оптимізацією зображення растра, що помітно прискорює його роботу по відношенню до решти методам, таким, як помірна, медіанна фільтрація, згладжування та ін.

Крім того, алгоритм міжкадрової фільтрації, на відміну від, скажімо, алгоритму лінійної фільтрації зі згладжувачим ядром, що не розмазує кордонів і контурів об'єктів, що є його важливою перевагою. Якість роботи алгоритму можна назвати цілком задовільним. При цьому швидкість оброблення приблизно в 4-5 разів перевищує швидкість роботи алгоритмів, що використовують локальні перетворення.

Алгоритми корекції яскравості і контрастності зображень було вирішено не використовувати, що дало додаткове прискорення роботи в 1,5-2 рази.

2.4.2 Фіксація руху

Після завершення етапу предобробки кадрів з метою підготовки їх до виявлення руху, можна приступити до виконання послідовності дій, що включає обчислення міжкадрової різниці, формування бінарної маски руху, виявлення руху.

При обчисленні міжкадрової різниці в якості одного з кадрів береться базовий кадр. Переваги такого підходу детально описані в другому розділі. Серед них: точність локалізації об'єкта на поточному кадрі, виявлення малорозмірних об'єктів, наявність ненульових елементів в масці руху не по контуру, а по всій площі об'єкта. Однак за це доводиться платити додатковими витратами на підготовку базового кадру (в межах 10%) і ускладненням алгоритму, що включає додавання зворотного зв'язку по рухомих об'єктів.

За значеннями міжкадрової різниці будується маска руху з використанням заданого користувачем порога. Елементи зі значеннями, меншими від порогу, обнуляються, інші стають одиницями в масці руху. У побудованій таким чином масці рухомих об'єктів залишаються чіткі сліди, які за формою досить сильно нагадують силуети вихідних об'єктів.

Щоб виділити рухомі об'єкти в масці руху, здійснюється послідовний обхід її пікселів з позначкою знайдених ненульових елементів. При знаходженні ненульового непоміченого елемента в масці, запускається процедура пошуку решти 8-зв'язкових з ним ненульових елементів за допомогою хвильового алгоритму. При цьому коригуються розміри прямокутника, який спочатку має нульову площу, розташовуючись в точці, в якій знаходиться вихідний піксель, а потім збільшується в розмірах слідом за включенням нових пікселів в групу. З точки зору математики прямокутник не може мати нульової площі, проте нам такий підхід зручний при реалізації алгоритму, тому що прямокутник в даному випадку завжди буде обмежувати групу пікселів, не виходячи за її межі, що дозволить уникнути помилок виходу за кордон масиву при роботі з об'єктами, що знаходяться у краї зображення.

Якщо відповідність знайдено, значить об'єкт з попереднього кадру був виявлений на поточному, і новий об'єкт в масив не заноситься. Замість цього відбувається лише оновлення параметрів вже наявного об'єкта. Коригується його поточний стан, розмір. Якщо відповідність не було знайдено, значить, об'єкт, ймовірно, зник за якою-небудь перешкодою. Такий об'єкт залишається в масиві об'єктів, однак позначається як зниклий. Тепер для всіх нововиявлених об'єктів буде обчислюватися їх схожість на цей зниклий об'єкт, поки або не буде знайдено схожість, що могло б означати, що об'єкт з'явився знову, поки не закінчиться інтервал часу, після якого схований об'єкт вважається зниклим і видаляється з масиву. Цей інтервал був обраний рівним 200 кадрів, чого цілком достатньо для безперервного трасування об'єкта, який зник навіть за великим перешкодою. По завершенні обробки об'єктів поточного кадру, всі зміни на поточному кадрі, крім тих, які викликані рухомими об'єктами, фіксуються в базовому кадрі. Це відбувається шляхом попиксельного оновлення областей базового кадру, в яких рухомі об'єкти не були виявлені, пікселями поточного кадру.

Тут необхідно сказати про просту корекцію, використану при оновленні базового кадру, яка дозволяє зменшити реакцію алгоритму на погойдування камери і скачки яскравості вихідного зображення, можливі при включенні / виключенні штучного освітлення, при зміні погоди.

Ідея полягає в тому, щоб коригувати пікселі базового кадру таким чином, щоб ця корекція мінімізувала різницю між значеннями пікселів поточного і базового кадру. Такий підхід дозволяє не реагувати на хитання камери і різкі скачки яскравості, пов'язані зі змінною освітлення, які нечасто тривають протягом великої кількості кадрів, і потім швидко гасяться в базовому кадрі за допомогою даного перетворення. Отже, алгоритм буде працювати за наступною схемою:

На схемі показана послідовність застосування операцій, спрямованих на виявлення рухомих об'єктів. Блок виділення контурів об'єктів не приведений. Малювання виконується після закінчення обробки даних поточного кадру, коли об'єкти виявлені, а всі їхні параметри записані в масив рухомих об'єктів.

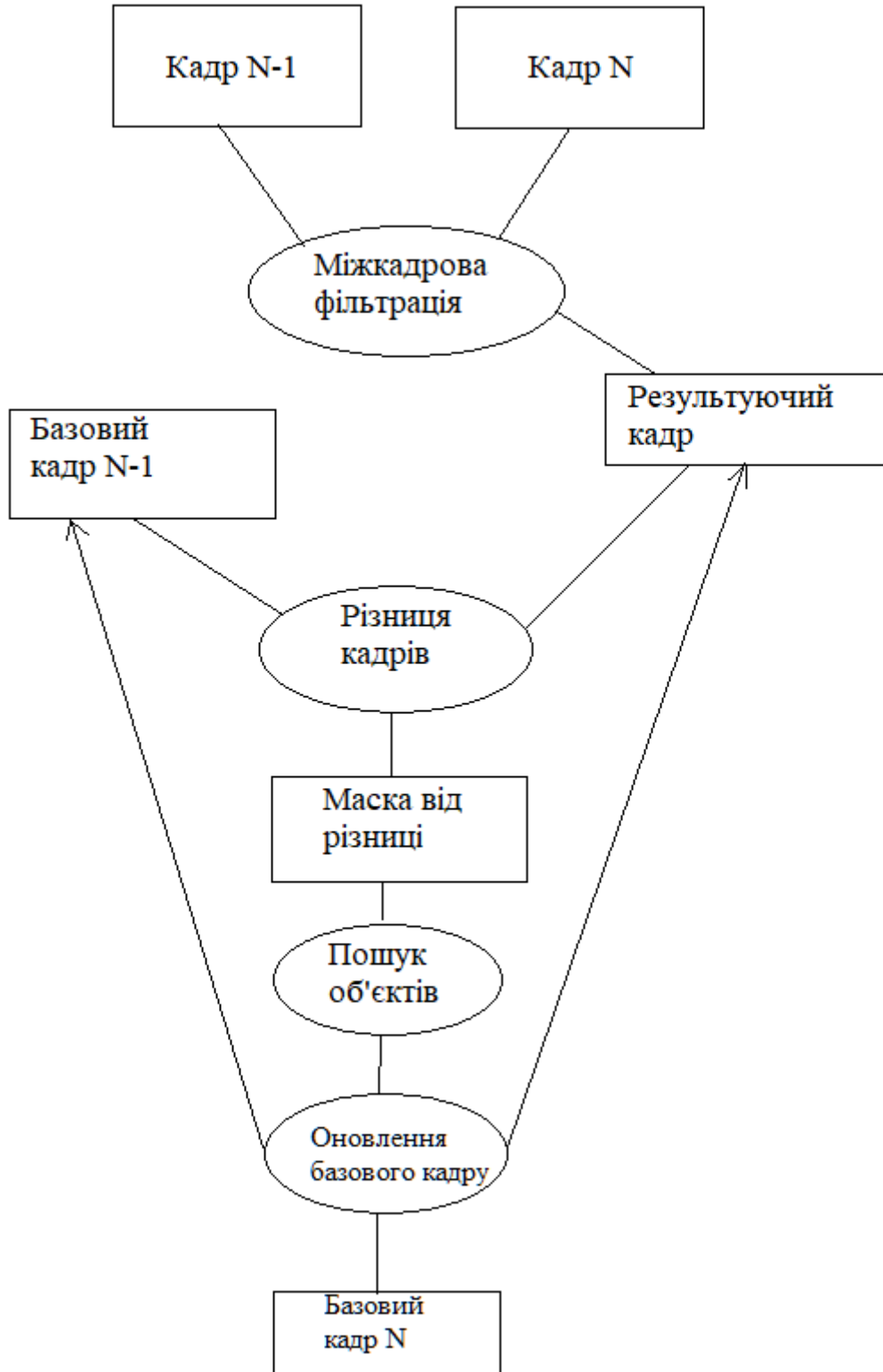


Рисунок 2.7 – Схема роботи алгоритму

3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ПРОЕКТУ

3.1 Огляд мови JAVA

Java – об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems». В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

«Oracle» надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License.

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім Java розроблялась як платформо-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням, що в порівнянні, наприклад, з C++ зменшує швидкість роботи програм. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування.

У мови Java є багато переваг перед іншими мовами програмування, що дозволяє вирішувати з його допомогою практично будь-які завдання.

Нижче перераховані основні переваги Java:

– Мова Java проста для вивчення. При розробці Java було приділено велику увагу простоті мови, тому програми на Java, в порівнянні з програмами на інших мовах, простіше писати, компілювати, налагоджувати і вивчати.

– Java – це об'єктно-орієнтована мова. Це дозволяє створювати модульні програми, вихідний код яких може використовуватися багаторазово.

– Мова Java не залежить від платформи. Одним з основних переваг мови Java є можливість перенесення програм з однієї системи в іншу. Оскільки програми на Java не залежать від платформи як на рівні вихідного коду, так і на довічнім рівні, їх можна запускати в різних системах.

Широкі можливості Java, простота застосування, незалежність від платформи і вбудовані функції захисту роблять цю мову програмування одним з кращих для створення додатків.

3.2 Огляд бібліотеки OpenCV

OpenCV – бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. Реалізована на C / C ++, також розробляється для Python, Java, Ruby, Matlab, Lua та інших мов. Може вільно використовуватися в академічних і комерційних цілях - поширюється в умовах ліцензії BSD.

Застосовується для затвердження загального стандартного інтерфейсу комп'ютерного зору для додатків в цій області. Для сприяння зростанню числа таких додатків і створення нових моделей використання РС.

Бібліотека містить понад 2500 оптимізованих алгоритмів, серед яких повний набір як класичних так і практичних алгоритмів машинного навчання і комп'ютерного зору. Алгоритми OpenCV застосовують у таких сферах:

- аналіз та обробка зображень;
- системи з розпізнавання обличчя;
- ідентифікації об'єктів;
- розпізнавання жестів на відео;
- відстежування переміщення камери;
- побудова 3D моделей об'єктів;

- створення 3D хмар точок зі стерео камер;
- трекінг відео.

Ця бібліотека була обрана для використання при розробці програмного продукту, так як задовольняє всім вимогам і задачам та значно спрощує розробку.

3.3 Програмна реалізація

В ході даної атестаційної роботи було розроблено програмний комплекс. Програма дозволяє відокремлювати динамічну складову в відео потоці. На основі аналізу даних відеоряду відбувається пошук рухомих об'єктів. Система розпізнавання руху, що покладено в основу програми, базується на власному алгоритмі, який порівнює два порядкових кадра відеоряду та знаходить рухомі об'єкти, які задовольняють встановленим налаштуванням.

Програма в реальному часі оброблює відео з камери спостереження у будинку. Розраховує низку характеристик геометричних та динамічних параметрів об'єкту, що рухається (наприклад, габарити, швидкість та інші). Результат системи розпізнавання дозволяє обрати контролюючу дію, наприклад надіслати власнику будівлі застереження за допомогою SMS повідомлення або електронної пошти.

Головне вікно програми має вигляд (рис.3.1):

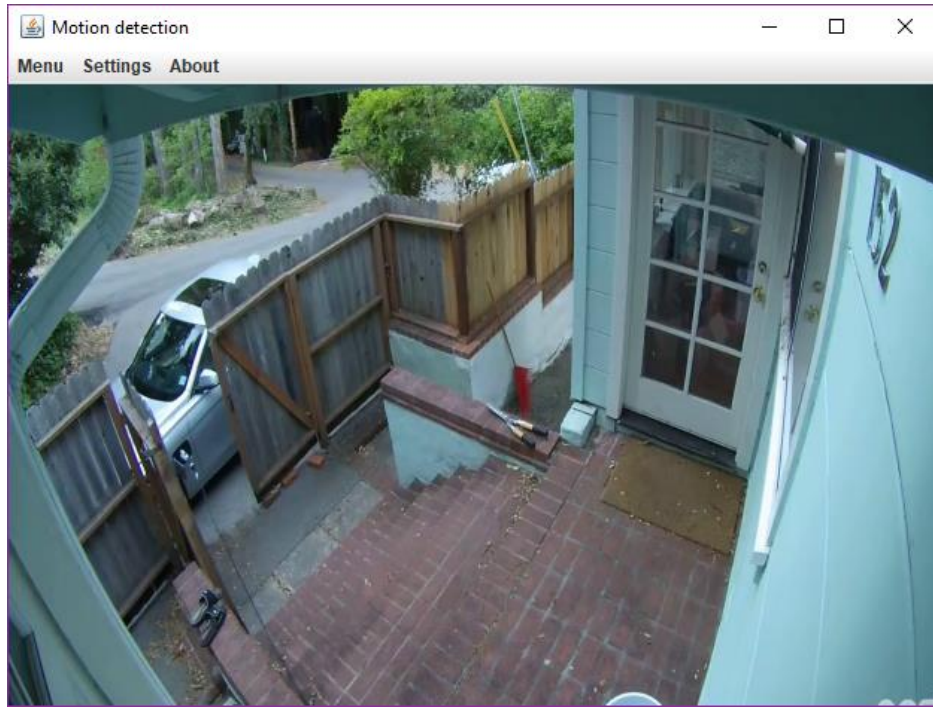


Рисунок 3.1 – Головне вікно програми

За допомогою головного меню даної програми є можливість переключатися між різними камерами (рис. 3.2).

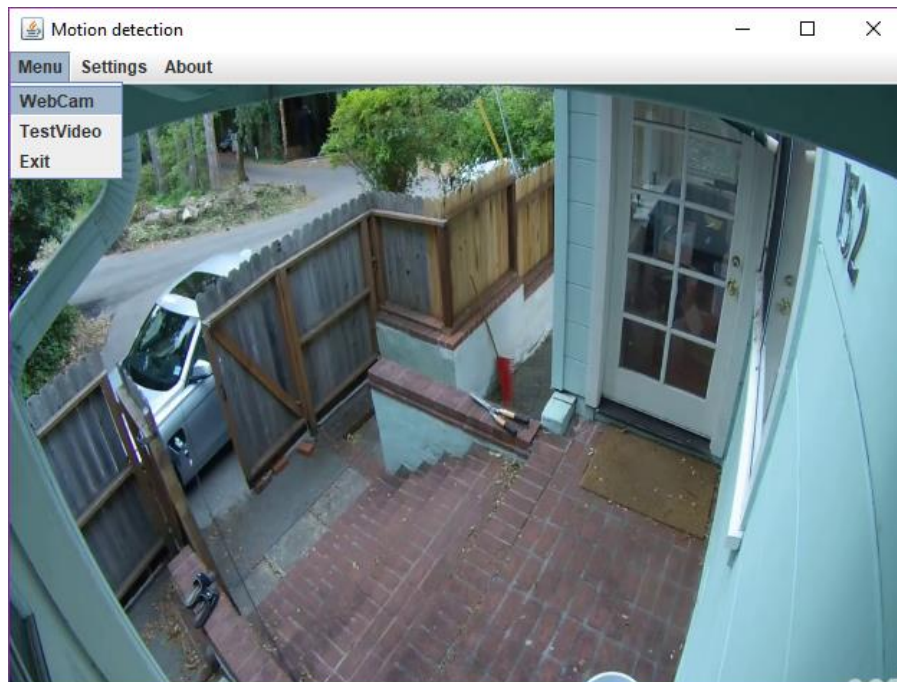


Рисунок 3.2 – Відчинене меню

Детектор руху працює в режимі реального часу. При виявленні руху, об'єкт виділяється спеціальною рамкою (рис. 3.3) та виводиться застережливе повідомлення на екран (рис. 3.4). Якщо активувати підключення електронної пошти, з'яляється можливість отримувати повідомлення про рух на електронну пошту (рис 3.5).

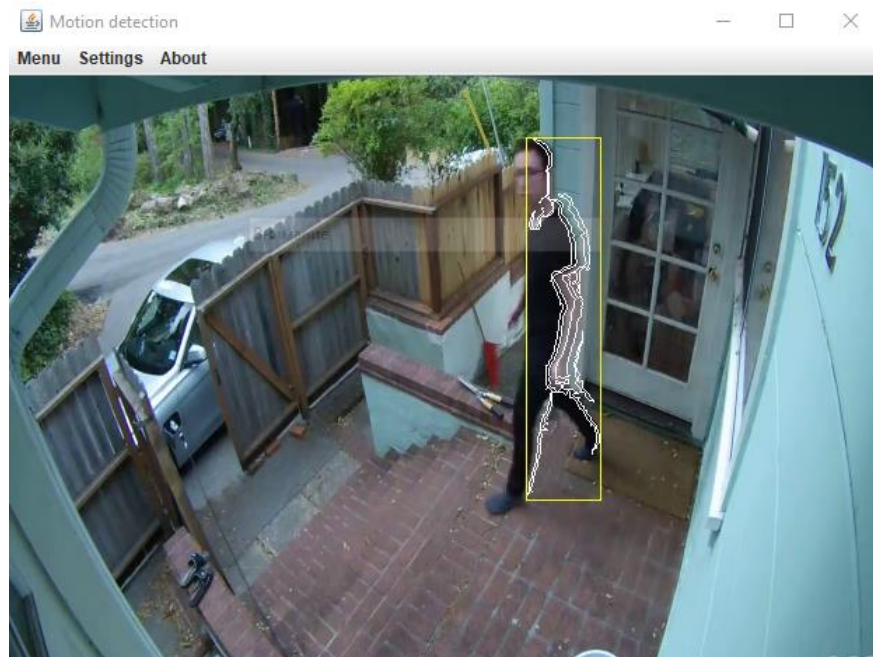


Рисунок 3.3 – Активований детектор руху

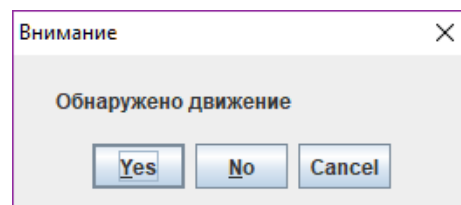


Рисунок 3.4 – Застережливе повідомлення

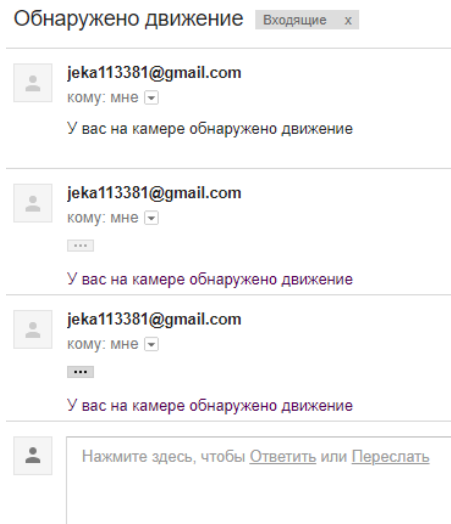


Рисунок 3.5 – Повідомлення про рух на електронній пошті

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.

4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкту, що мають вплив на персонал

У даному дипломному проєкті розробляється програмне забезпечення.

Розроблене програмне забезпечення орієнтоване на роботу з персональним комп'ютером. Експлуатовані для вирішення внутрішньовиробничих завдань ПЕОМ типу IBM PC мають наступні характеристики:

споживана потужність	220 Вт;
робоча напруга	220 В;
напруга джерел живлення	+12 В; - 12 В; +5 В;
робоча частота	50 Гц.

Виходячи з приведених характеристик, вочевидь, що для людини існує небезпека поразки електричним струмом, унаслідок недбалого поводження з комп'ютером і порушення правил експлуатації, залишення частин ПЕОМ, що знаходяться під напругою, відкритими або знятих для ремонту вузлів.

Відповідно до [16] до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

При роботі на ПЕОМ користувач піддається ряду потенційних небезпек. Унаслідок недотримання правил техніки безпеки при роботі з машиною (невиконання огляду відкритих частин ПЕОМ, що знаходяться під напругою або знятих для ремонту вузлів) для користувача існує небезпека поразки електричним струмом.

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;
- блоки ПЕОМ і друку, що знаходяться в ремонті.

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння мала, оскільки цей вид випромінювання поглинається речовиною екрану. Проте велику увагу слід приділяти біологічним ефектам низькочастотних електромагнітних полів (аж до порушення ДНК).

Відповідно до [17], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

- підвищене значення напруги в електричному ланцюзі, замикання якої може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювань;
- підвищена або знижена температура повітря робочої зони;
- підвищений або знижений рух повітря;
- підвищена або знижена вологість повітря;
- відсутність або недостатність природного світла;
- підвищена пульсація світлового потоку;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційні навантаження;
- монотонність праці.

4.2 Заходи щодо техніки безпеки

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка посилюється тим, що органи

чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм чинить на нього складну дію, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної (роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Тяжкість поразки людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- роду і частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.

Розроблений дипломний проект передбачає наступні технічні способи і засоби, що застерігають людину від ураження електричним струмом:

- заземлення електроустановок;
- занулення;
- захисне відключення;
- електричне розділення мережі;
- використання малої напруги;
- ізоляція частин, що проводять струм;
- огорожа електроустановок.

Занулення зменшує напругу дотику і обмежує години, протягом яких людина, ткнувшись до корпусу, може потрапити під дію напруги.

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_k = \frac{U_\phi}{Z_\Pi + \frac{Z_T}{3}}, \quad (4.1)$$

де U_ϕ - номінальна фазна напруга мережі, В;

Z_Π - повний опір петлі, створене фазними і нульовими дротами, Ом;

Z_T - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4 [18]: $Z_T/3 = 0,1$ Ом.

Для провідників і жил кабелю для розрахунку повного опору петлі використовуємо формулу(4.2.) :

$$Z_\Pi = \sqrt{R_\Pi^2 + X_\Pi^2}, \quad (4.2)$$

де $R_\Pi = R_\phi + R_0$ - сумарний активний опір фазного R_ϕ і нульового R_0 дротів, Ом;

X_Π - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту $S = 2.5$ мм, тоді згідно таблицям 5 і 6 [18], має такий опір:

$$X_\Pi = 0,11 \text{ Ом};$$

$$R_\phi = 7,55 \text{ Ом};$$

$$R_0 = 7,55 \text{ Ом}.$$

$$\text{Отже, } R_\Pi = 7,55 + 7,55 = 15,1 \text{ Ом}.$$

Тоді по формулі (4.2) знаходимо повний опір петлі :

$$Z_\Pi = \sqrt{15,1^2 + 0,11^2} \approx 15,1 \text{ (Ом)}.$$

Струм однофазного короткого замикання рівний:

$$I_k = \frac{220}{15,1 + 0,1} = 14,47 \text{ (А)}.$$

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$I_k \geq k * I_n, \quad (4.3)$$

де I_n - номінальний струм спрацьовування плавкої вставки, А;

k - коефіцієнт кратності нелінійного струму I_n , А.

Коефіцієнт кратності нелінійного струму I_n розраховується по формулі (4.4.) :

$$I_n = P / U, \quad (4.4)$$

де $P = 220$ Вт - споживана потужність;

$U = 220$ В - робоча напруга;

$k = 3$ А - для плавких вставок.

Отже, $I_n = 220 / 220 = 1$ А.

Підставивши значення у вираз (4.3), одержимо:

$$14,47 > 3 * 1.$$

Таким чином, доведено, що апарат забезпечить спрацьовування(і захист) при підвищенні номінального струму.

4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Вимоги до виробничих приміщень встановлюються [23], ДБН, відповідними ГОСТами і ОСТАми з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

Підвищення працездатності людини і збереження її здоров'я забезпечується стабільними метеорологічними умовами.

Мікроклімат виробничих приміщень визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і потовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

Відповідно до [16] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні. За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

У приміщенні для виконання робіт операторського типу (категорія 1а), пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (табл. 4.1).

Таблиця 4.1 - Санітарні норми мікроклімату робочої зони приміщень для робіт категорії 1а.

Пора року	Температура, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодна	22...24	40...60	0,1
Тепло	23...25	40...60	0,1

У приміщенні, де знаходиться ПЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (з пристроєм вентиляційних каналів в перекриттях будівлі і вертикальних шахт) й установленого промислового кондиціонера фірми Mitsubishi, який дозволяє вирішити переважну більшість завдань по створінню та підтримці необхідних параметрів повітряного середовища. Цей метод забезпечує приток потрібної

кількості свіжого повітря, визначеного в ДБН (30 м³ в годину на одного працівника).

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму, використовують наступні методи. При покупці принтера слід вибирати найбільш шумозахисні матричні принтери або з великою швидкістю роботи(струменеві, лазерні). Рекомендується принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання(під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повсті, поролону, пеноплону).

При роботі на ПЕОМ, проектом передбачені наступні методи захисту від електромагнітного випромінювання : обмеження часом, відстанню, властивостями екрану.

Обмеження годині роботи на ПЕОМ складає 3,5-4,5 години. Захист відстанню передбачає розміщення монітора на відстані 0,4-0,5 м від оператора. Передбачений монітор 20" TFT, Samsung 2043BW відповідає вимогам стандарту ТСО'03.

ТСО'03 пред'являє жорсткі вимоги в таких областях: ергономіка (фізична, візуальна і зручність користування), енергія, випромінювання (електричних і магнітних полів), навколишнє середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам [19].

Для зниження стомлюваності та підвищення продуктивності праці обслуговуючого персоналу в колірній композиції інтер'єру приміщень для ПЕОМ дипломним проектом пропонується використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

У проекті передбачається використання сумісного освітлення. У світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Як штучне освітлення необхідно використовувати штучне робоче загальне освітлення. Для загального освітлення необхідно використовувати люмінесцентні лампи. Вони володіють наступними перевагами: високою світловою віддачею, тривалим терміном служби, хоча мають і недоліки: високу пульсацію світлового потоку.

При експлуатації ПЕОМ виробляється зорова робота. Відповідно до [23] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці(E_n) при загальному освітленні рівна 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛП02П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

Виконаємо розрахунок кількості світильників в робочому приміщенні завдовжки $a=12$ м, шириною $b=10$ м, заввишки $z=4$ м, використовуючи формулу (4.5) розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.5)$$

де F - світловий потік = 3120 лм;

E - максимально допустима освітленість робочих поверхонь = 200 лк;

S - площа підлоги = 120 м²;

Z - поправочний коефіцієнт світильника = 1,2;

k - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників = 1,5;

n - кількість світильників;

U - коефіцієнт використання освітлювальної установки = 0,6;

M - кількість ламп у світильнику = 2.

З формули (4.5) виразимо n (4.6) і визначимо кількість світильників для даного приміщення:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.6)$$

Отже, $n = (200 \cdot 120 \cdot 1,2 \cdot 1,5) / (3120 \cdot 0,6 \cdot 2) = 12$.

Виходячи з цього, рекомендується використовувати 12 світильників. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. 4.1.

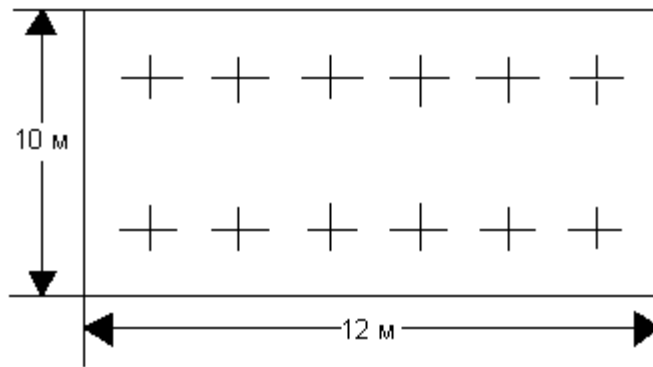


Рисунок 4.1 - Схема розташування світильників

4.4 Рекомендації по пожежній безпеці

Пожежі в приміщеннях, де встановлена обчислювальна техніка, представляють небезпеку для життя людини. Пожежі також пов'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що у свою чергу спричиняє за собою порушення ходу технологічного процесу.

Пожежа може виникнути при наявності горючої речовини та внесення джерела запалювання в горюче середовище. Пальними матеріалами в приміщеннях, де розташовані ПЕОМ, є:

– поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 °С ;

– полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, температура самозаймання 530 °С, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;

– стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;

– пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;

– деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 °С, схильна до самозаймання.

Згідно [22] приміщення відносяться до категорії В (пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір усередині приміщення відноситься до вогнебезпечної зони класу П - Па (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

- іскри при замиканні і розмиканні ланцюгів;
- іскри і дуги коротких замикань;
- перегриви від тривалого перевантаження і наявності перехідного опору.

Продуктами згорання, що виділяються при пожежі, є: оксид вуглецю, сірчистий газ, оксид азоту, синильна кислота, акропеїн, фосген, хлор та ін. При горінні пластмас, окрім звичайних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол та ін., що шкідливо впливають на організм людини.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигаза з коробкою марки В(жовта).

Пожежна безпека об'єктів народного господарства регламентується [22] і забезпечується системами запобігання пожежам і протипожежному захисту. Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

Зменшити горюче навантаження не представляється можливим, тому проектом передбачається застосувати наступні способи і їх комбінації для запобігання утворенню(внесення) джерел запалення :

- застосування устаткування, що задовольняє вимогам електростатичної безпеки;
- застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;
- виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;
- підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту з палим середовищем, нижче гранично допустимої, становить 80% якнайменшої температури самозаймання пального.
- заміна небезпечних технологічних операцій більш безпечними;
- ізольоване розташування небезпечних технологічних установок і устаткування;
- зменшення кількості палих і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;
- запобігання можливості утворення палих сумішей на лінії, вентиляційних системах і ін.;
- механізація, автоматизація та справність(потокова) виробництва;

- суворе дотримання стандартів і точне виконання встановленого технологічного режиму;
- запобігання можливості появи в небезпечних місцях джерел запалення;
- запобігання розповсюдженню пожеж і вибухів;
- використання устаткування і пристроїв, при роботі яких не виникає джерел запалення;
- виконання вимог сумісного зберігання речовин і матеріалів;
- наявність громовідводу;
- ліквідація можливості самозаймання речовин і матеріалів .

Для запобігання пожежі в обчислювальних центрах проектом пропонується виконання наступних вимог :

- електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолодження і кондиціонування;
- система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;
- робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

Для зниження пожежної небезпеки в приміщеннях використовуються первинні засоби гасіння пожеж, а також система автоматичної пожежної сигналізації, яка дозволяє знайти початкову стадію загоряння, швидко і точно оповістити службу пожежної охорони про час і місце виникнення пожежі.

Відповідно до правил пожежної безпеки для промислових підприємств приміщення категорії В підлягають устаткуванню системами автоматичної пожежної сигналізації. Проектом передбачається застосування датчика типу ІДФ - 1(димовий фотоелектричний датчик), оскільки специфікою пожеж

обчислювальної техніки і радіоапаратури є, в першу чергу, виділення диму, а потім - підвищення температури.

При виникненні пожежі в робочому приміщенні обслуговуючий персонал зобов'язаний негайно вжити заходи по ліквідації пожежі. Для ліквідації пожежі використовують вогнегасники (хімічно-пінні, пінні для повітря ОП-5, ОП-6, ОП-9, вуглекислотні ОУ-5), пісок, пожежний інвентар (сокири, ломи, багри, шерстяну або азбестову ковдри). Як засіб індивідуального захисту проектом передбачається використання промислового протигаза з маскою, фільтруючої коробки В.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

ВИСНОВКИ

Метою цієї роботи була розробка детектора рухомих об'єктів, здатного в реальному часі виявляти і відстежувати рух об'єктів у широкому діапазоні навколишніх умов. В рамках дипломної роботи був розроблений і реалізований метод міжкадрової фільтрації, був спроектований та реалізований програмний охоронний комплекс.

Програмний комплекс в реальному часі оброблює відео з камери спостереження у будинку. Розраховує низку характеристик геометричних та динамічних параметрів об'єкту, що рухається (наприклад, габарити, швидкість та інші). Результат системи розпізнавання дозволяє обрати контролюючу дію, наприклад надіслати власнику будівлі застереження за допомогою SMS повідомлення або електронної пошти.

Крім того, що вже реалізовано, існує ряд можливостей модернізації програми з метою розширення її функціональності і поліпшення характеристик, які можуть бути корисними в охоронній системі. Крім виявлення, програма могла б класифікувати виявлені об'єкти або, принаймні, давати відповідь на питання, чи виявлявся даний об'єкт раніше. Є можливість розширення програмного комплексу додатками на ios і Android. Такими можуть бути плани розвитку даного програмного продукту на майбутнє.

Отже, запропонований охоронний програмний комплекс полегшить спостереження за будинком, зробить його більш ефективним за допомогою відправки повідомлень користувачу та дозволить заощадити на покупці детекторів руху, через можливість використовувати будь-яку відеокамеру.

У розділі «Охорона праці» виконано аналіз потенційних небезпек при роботі із засобами обчислювальної техніки і механізмами, розроблені заходи щодо техніки безпеки, заходи, які забезпечують виробничу санітарію і гігієну праці, розраховане штучне освітлення, виконані рекомендації по пожежній безпеці.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

- 1) Fisher, R. CVOnline: Motion and time sequence Analysis [Електронний ресурс]. – 2002. – Режим доступу до ресурсу: <http://homepages.inf.ed.ac.uk/rbf/CVonline/motion.htm#chgdet>, вільний.
- 2) Morris, R. An Integrated Traffic and Pedestrian Vision System [Електронний ресурс]. – 1998. – Режим доступу до ресурсу: <http://www.scs.leeds.ac.uk/imv/index.html>, вільний.
- 3) Вежневцев, В. Введення в Computer Vision [Електронний ресурс]. - 2003. – Режим доступу до ресурсу: http://cgm.graphicon.ru:8080/issue0/cv_intro/index.html, вільний.
- 4) Davies D., Palmer P., Mirmehdi M. Detection and Tracking of Very Small Low Contrast Ob-jects [Електронний ресурс]. – 1994. – Режим доступу до ресурсу: <http://www.bmva.ac.uk/bmvc/1998/pdf/p135.pdf>, вільний.
- 5) Сойфер, В.А. Комп'ютерна обробка зображень. Частина 2. Методи і алгоритми. Соросівський освітній журнал, №3, 1996, с. 110 – 121.
- 6) Fisher, B., Perkins S., Walker A. HyperMedia Image Processing Reference [Електронний ресурс]. – 1994. – Режим доступу до ресурсу: http://www.cee.hw.ac.uk/hipr/html/hipr_top.html, вільний.
- 7) Young S., Forshaw M., Hodgetts, M. Change detection for perimeter surveillance. - University College London Image Processing Group Internal report [Електронний ресурс]. – 1998. – Режим доступу до ресурсу: <http://ipga.phys.ucl.ac.uk/reports/index.html>, вільний.
- 8) Young S., Forshaw M., Hodgetts, M. Image comparison methods for perimeter surveillance. – University College London Image Processing Group [Електронний ресурс]. – 1999. – Режим доступу до ресурсу: <http://ipga.phys.ucl.ac.uk/reports/index.html>, вільний.
- 9) Young I., Gerbrands J., van Vliet, L. Image Processing Fundamentals [Електронний ресурс]. – 1999. – Режим доступу до ресурсу: <http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip.html>, вільний.

10) Motion Detection and Segmentation. R.Brooks, C.Breazeal, M.Marjanovic, B.Scassellati, [Електронний ресурс]. – 1998. - Режим доступу до ресурсу: http://www.ai.mit.edu/projects/cog/VisionSystem/motion_detection.html, вільний.

11) Страуструп, Б. Язык программирования C++. Часть 1 [Текст] / Б. Страуструп; перевод с англ. – К.: ДияСофт, 1993. – 264 с.

12) Страуструп, Б. Язык программирования C++. Часть 2 [Текст] / Б. Страуструп; перевод с англ. – К.: ДияСофт, 1993. – 296 с.

13) Фор, А. Восприятие и распознавание образов [Текст] / А. Фор; перевод с фр. – М.: Машиностроение, 1989. – 271 с.

14) Форсайт, Д.А. Компьютерное зрение. Современный подход. [Текст] Д.А. Форсайт, Д. Понс; перевод с англ. – К.: Вильямс, 2004. – 928 с.

15) Шапиро, Л. Компьютерное зрение [Текст] / Л. Шапиро, Дж. Стокман; перевод с англ. – М.: БИНОМ. Лаборатория знаний, 2006. – 752 с.

16) ДСН 3.3.6.042-99 Державні санітарні норми мікроклімату виробничих приміщень. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/va042282-99

17) НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/z0508-18

18) ДСТУ 7237:2011 Національний стандарт України. Система стандартів безпеки праці. Електробезпека. Загальні вимоги та

19) номенклатура видів захисту. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/ru/v0037831-11

20) ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/v0007282-98

- 21) ДБН В.2.5-67:2013. Опалення вентиляція та кондиціонування. Режим доступу: www. URL: <https://zakon.rada.gov.ua/rada/show/v0024858-13>
- 22) ДБН В.2.5-28-2006. Природне і штучне освітлення. Режим доступу: www. URL: <https://zakon.rada.gov.ua/rada/show/v0168667-06>
- 23) ДСТУ Б В.1.1-36-2016. Визначення категорії приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою. Режим доступу: www. URL: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=65419
- 24) ДСП 173-96. Державні санітарні правила планування та забудови населених пунктів. Режим доступу: www. URL: <https://zakon.rada.gov.ua/laws/show/z0379-96>
- 25) Симметрон. Электронные компоненты. Каталог 2002, 2002г. – 192с.

ДОДАТОК 1. ЕЛЕКТРОННІ ПЛАКАТИ

Міністерство освіти і науки України
Східноукраїнський національний університет імені
Володимира Даля

Дипломна робота

«Інформаційна система детектору рухомих об'єктів»

Виконав:
ст.гр. КІ-15з
Джафаров В.В.

Керівник:
проф. Рязанцев О.І.

2019 р.

Слайд 2

Актуальність теми

Існує багато завдань, в яких зображення розглядаються як джерело інформації, на основі якої необхідно винести деяке рішення.

У даній роботі було вирішено реалізувати охоронну систему стеження. Актуальність роботи підтверджується статистикою зросту інтересу до розумних технологій та можливостей автоматизації повсякденних дій.

Очевидний приклад розумного відеоспостереження – детекція руху. Один детектор руху здатний замінити відразу кілька операторів відеоспостереження.

Постановка задачі

Розробка детектора руху полягає в дослідженні і застосуванні методів обробки (фільтрації) растрових зображень і виявлення на них рухомих об'єктів, таким чином, щоб забезпечити виконання наступних вимог:

- можливість роботи детектора в реальному часі;
- стійку роботу в широкому діапазоні зовнішніх умов;
- мінімальна кількість помилкових спрацьовувань на шум;
- мінімальна кількість спрацьовувань на об'єкти, які не становлять небезпечності для охоронюваного периметра / об'єкта (хитні дерева, листя, інша рослинність та ін.).

Якість роботи розробленого детектора перевіряється на відеороликах, знятих різними цифровими відеокамерами в широкому діапазоні зовнішніх умов.

Предобробка

- На етапі попередньої обробки було необхідно підготувати відеозображення до виявлення на них рухомих об'єктів. Можна вважати, що попередня обробка проведена якісно, якщо будь-які два відеокадра, отримані з невеликим інтервалом від однієї і тієї ж камери в умовах відсутності руху в зоні видимості камери є максимально схожими один на одного.

Було вирішено застосувати алгоритм міжкадрової фільтрації

Перевагою алгоритму є його швидкодія. Він працює за один прохід, не вимагаючи додаткової обробки і накопичення інформації.



Фіксація руху

Для фіксації руху будемо використовувати міжкадрову різницю з базовим кадром.

$$R_{res}^i = |R_1^i - R_2^i|$$

$$G_{res}^i = |G_1^i - G_2^i|$$

$$B_{res}^i = |B_1^i - B_2^i|$$

$$p^i = (R_{res}^i + G_{res}^i + B_{res}^i) \cdot 3$$

Середнє значення порівнюється із заданим порогом. В результаті порівняння формується двійкова маска руху

$$m^i = \begin{cases} 0, p^i \leq T \\ 1, p^i \geq T \end{cases}$$

Аналіз знайдених контурів

Виділення об'єктів відбувається при послідовному обході пікселів маски руху. Якщо в масці руху виявлена одиниця, запускається процедура пошуку пікселів, зв'язкових з даними, яким також відповідають одиниці в масці руху. Знайдені пікселі позначаються, щоб уникнути повторного їх виявлення.

Після цього проводиться аналіз площі знайденого прямокутника і виключаються занадто маленькі або великі об'єкти

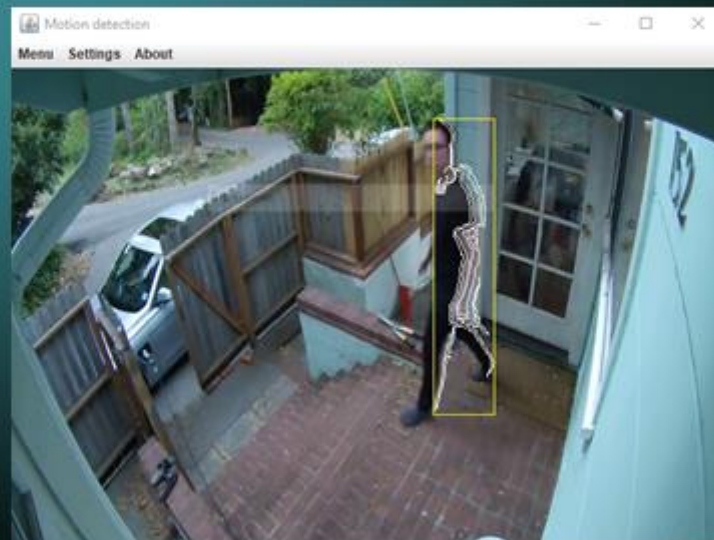
Алгоритм програми

Слайд 9



Результат роботи програми

Слайд 10



Висновки

Слайд 11

В рамках атестаційної роботи був розроблений і реалізований метод міжкадрової фільтрації, був спроектований та реалізований програмний охоронний комплекс.

Крім того, що вже реалізовано, існує ряд можливостей модернізації програми з метою розширення її функціональності і поліпшення характеристик, які можуть бути корисними в охоронній системі. Крім виявлення, програма могла б класифікувати виявлені об'єкти або, принаймні, давати відповідь на питання, чи виявлявся даний об'єкт раніше.

Є можливість розширення програмного комплексу додатками на ios і Android. Такими можуть бути плани розвитку даного програмного продукту на майбутнє.