МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ, МОЛОДІ ТА СПОРТУ
ЗАПОРІЗЬКА ДЕРЖАВНА ІНЖЕНЕРНА АКАДЕМІЯ

Романенко Юрій Олександрович

УДК 004.9.89

РОЗРОБКА ТА ДОСЛІДЖЕННЯ ІГРОВИХ ПЛАТФОРМ ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ НА ПЛАТФОРМІ IOS МОВОЮ SWIFT

8.05010301 Програмне забезпечення систем

АВТОРЕФЕРАТ
дипломної роботи магістра

Запоріжжя – 2016

Кваліфікаційною роботою магістра є рукопис

Робота виконана на кафедрі програмного забезпечення автоматизованих систем Запорізької державної інженерної академії

Науковий керівник: доцент, кандидат фізико-математичних наук
Попівщий Василь Іванович,
доцент кафедри програмного забезпечення автоматизованих систем Запорізької державної інженерної академії

Офіційний рецензент: Орешкін Юрій Ігорович,
Начальник управління інраструктурних систем, ВАТ «Запоріжсталь»

Захист відбудеться «14» січня 2016 р. о $9^{\underline{00}}$ на засіданні Державної екзаменаційної комісії Запорізької державної інженерної академії за адресою:
690006 м. Запоріжжя, пр. Леніна 226, адміністративний корпус, ауд. 40.

1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність роботи

Ще десять років тому, управління будь-яким пристроєм за допомогою тіла без дотику до нього, було важко уявити. Зараз з’являється все більше технологій, пристроїв, бібліотек та програмного забезпечення для розпізнавання жестів та керування ними.

Жестами можна керувати комп’ютером, телевізором, ігровою консоллю та навіть автомобілем.

.

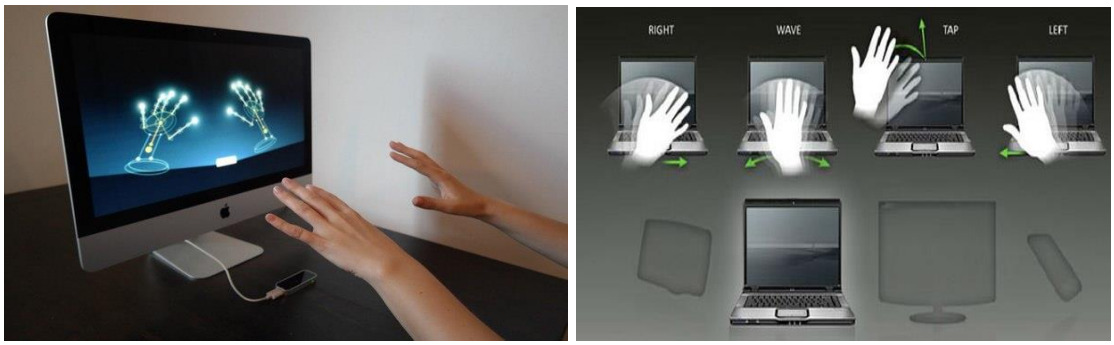


Рис. 1. Приклад управління комп’ютером.

Рис. 2. Приклад управління телевізором.

Рис. 3. Приклад управління ігровою консоллю.

Рис. 4. Приклад управління автомобілем.

І надалі таких пристроїв буде з’являтися все більше і більше, тому це одна з актуальніших тем у IT індустрії.

Ціль роботи.

Необхідно розробити застосунок, який розпізнає рух об’єктів перед фронтальною камерою мобільного пристрою за допомогою бібліотеки OpenCV під iOS.

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

Бібліотека розроблена Intel і нині підтримується Willow Garage та Itseez. Сирцевий код бібліотеки написаний мовою C++ і поширюється під ліцензією BSD. Біндинги підготовлені для різних мов програмування, таких як Python, Java, Ruby, Matlab, Lua та інших. Може вільно використовуватися в академічних та комерційних цілях.

Основні задачі дослідження.

Узагальнення і систематизація існуючої інформації про ігрові платформи для операційної системи iOS. Також інформації про використання різних мов програмування в тому числі і Swift для створення ігрових застосунків.

Розробка алгоритму для розпізнавання жестів рук за допомогою фронтальної камери пристрою. Пошук потрібних математичних алгоритмів для реалізації розпізнавання.

Пошук потрібної для реалізації алгоритму бібліотеки роботи з зображенням. Підключення і використання бібліотеки OpenCV під платформу iOS.

Розробка гри для демонстрації роботи механізму розпізнавання жестів.

Збірка і тестування всіх компонентів системи у один застосунок.

Наукова новизна.

Розроблений алгоритм розпізнавання жестів оснований на відстеження рухомих зон на зображення з фронтальної камери пристроїв на операційній системі iOS.

Створено аналог популярної гри «2048» на мові програмування Swift.

Використано бібліотеку для обробки зображення OpenCV, створену на C++. Для використання цієї бібліотеки застосунок має компонент реалізований на C++.

Для роботи всіх компонентів системи було використано **три** мови програмування: C++, Objective-C та Swift.

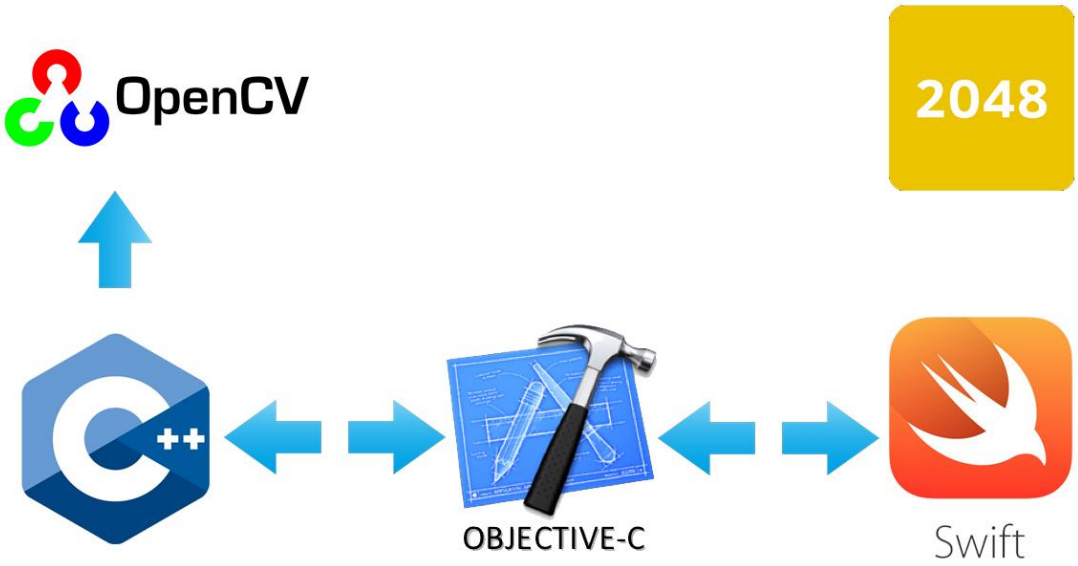


Рис. 5. Діаграма компонентів системи.

Розроблений ігровий застосунок, який без додаткових пристроїв та сенсорів дозволяє грати за допомогою жестів рук, без дотиків до екрану пристрою.

Практична значимість

Розроблений застосунок з високою точністю розпізнає жести рук перед пристроєм за допомогою фронтальної камери. Розроблений алгоритм можливо застосувати у будь-якому застосунку на будь-якій платформі. Від ігор до утиліт і управління операційною системою.

Рис. 6. Приклад управління планшетом жестами.

Наприклад алгоритм можна використати для застосунку «Галерея». Де жестами руки можна переглядати картинки, запускати відео, видаляти непотрібні фото та відео.

Апробація роботи

Основні положення й результати досліджень повідомлені й обговорені на XIV науково-технічної конференції студентів, магістрантів, аспірантів і викладачів ЗДІА «Електроніка, Автоматизовані системи й сучасні інформаційні технології» 13-17 квітня 2015 р. (м. Запоріжжя).

Публікації.

За результатами досліджень опублікована друкована робота в періодичних виданнях і працях наукових конференцій.

Основні положення й результати досліджень, а так само всі матеріали XIV науково-технічної конференції студентів, магістрантів, аспірантів і викладачів ЗДІА «МАТЕРІАЛИ XX НАУКОВО-ТЕХНІЧНОЇКОНФЕРЕНЦІЇ СТУДЕНТІВ, МАГІСТРАНТІВ,АСПІРАНТІВ І ВИКЛАДАЧІВ ЗДІА 2014-2015 н.р.» Запоріжжя: Видавництво ЗДІА, 2015.

Романенко Ю.О., магістрант гр. СП-14м, Попівщий В.Ш., доц. - науковий керівник.

«Вплив мови програмування SWIFT на IT-індустрію» (ст. 39)

Структура й об'єм роботи.

Кваліфікаційна дипломна робота магістра складається із введення, 4 глав і висновків, списку використаних джерел з 26 найменувань і додатка. Робота містить 106 сторінок тексту, 14 розрахункових формул, 37 малюнків, 6 таблиць.

ЗМІСТ РОБОТИ

У введенні описана мова програмування Swift, її історія, переваги та недоліки. Описується архітектура та походження мови, принципи роботи та використання мови. Показані темпи розвитку мови та особливості мови. Наводяться приклади синтаксу.

У першому розділі розкривається актуальність теми та огляд існуючих рішень. Наприклад, о Один із самих відомих аналогічних проектів для розпізнавання жестів – Leap Motion – технологія, основана на захопленні руху, створена для людино- машинної взаємодії.

Рис. 7. Фото пристрою The Leap.

The Leap - це невеликий USB-пристрій, розроблений для столу користувачів, робочою частиною розташовується вгору, тим самим створюючи 3D-область взаємодії об'ємом близько 227 дециметрів кубічних (тобто уявному кубі зі стороною 61 см). Як представлено на відеозаписах, всередині цієї області The Leap відстежує рух пальців і рук, олівців, ручок, паличок для їжі з великою точністю.

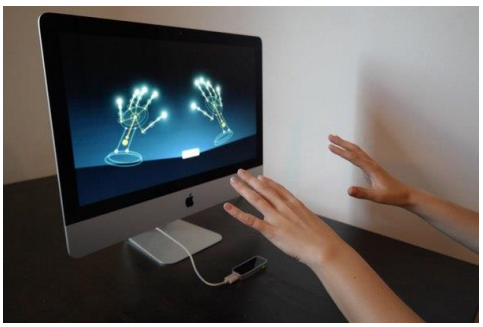


Рис. 8. Фото роботи з пристроєм The Leap.

Ще одне відоме рішення - технологія Intel® RealSence. Intel® RealSense, раніше відомий як Intel Perceptual Computing є платформою для реалізації методів взаємодії людини з комп'ютером за допомогою жестів. Вона складається з серії 3D камер разом з простою у використанні бібліотекою машиного сприйняття, що спрощує підтримку камер для розробників програмного забезпечення.

Рис. 9. Фото пристрою від Intel.

Особливості:

- Відстеження кількох осіб.
- Ідентифікація рис обличчя, очей, роту та носу
- Відстеження до 10 одночасних пальців, 8 жестів

У другому розділі викладені основні теоретичні підходи до розпізнавання жестів. Наприклад такі як застосування жестів рук при людино-машинному інтерфейсі.

Перспективним напрямком розвитку інформаційних технологій є розробка нових способів забезпечення інтерфейсу людина-машина. Перед розробниками подібних інтерфейсів ставиться задача використання природних для людини способів спілкування з комп’ютерами. Враховуючи усі можливі перешкоди та наявність шумів в оточуючому середовищі, перевага надається системам на основі комп’ютерного зору. Особливо перспективними для по-

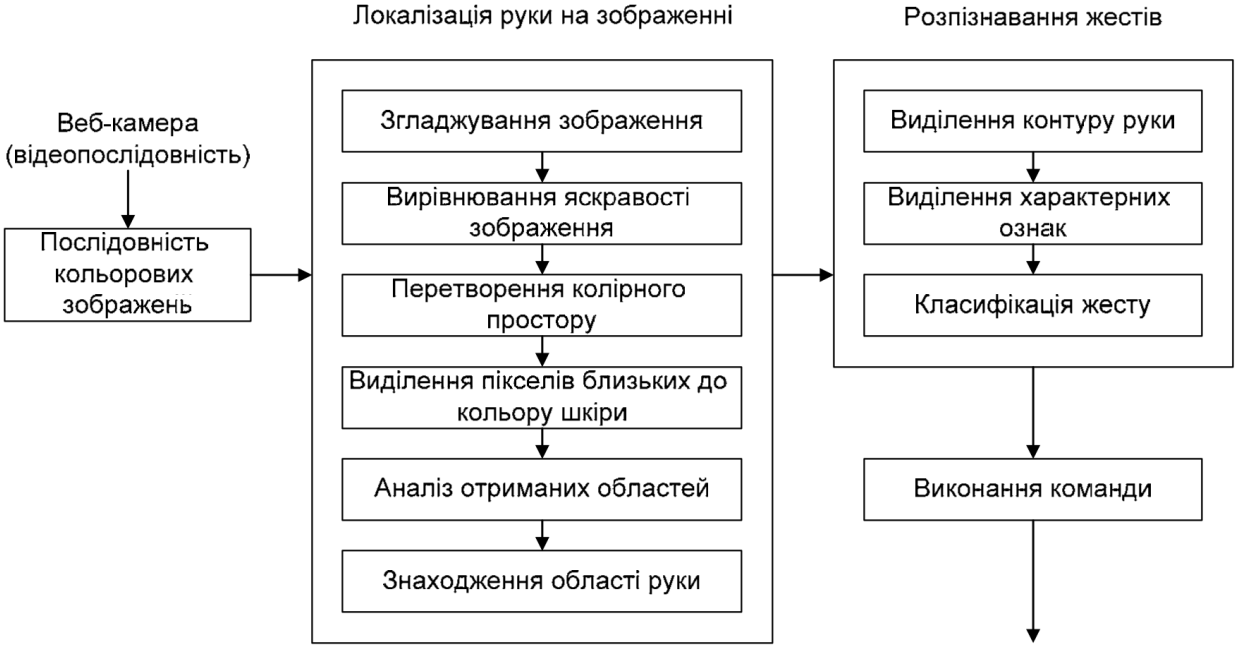
будови інтерфейсів управління програмним та апаратним забезпеченням комп'ютерів є жести. Перевага такого інтерфейсу полягає у тому, що жести дозволяють розширити можливості інтерфейсу для людей з вадами слуху і мови, та забезпечити дистанційне управління різними побутовими пристроями, хоча при їх створенні існує ряд не повністю вирішених задач, які розглядаються у статті.

Розглянемо основні етапи процесу розпізнавання жестів (рис. 10). Для реалізації основних функцій управління система управління жестами (СУЖ) може включати набір з 10 жестів (табл. 1).

Робота системи містить ряд основних етапів:

− захоплення зображення - здійснюється за допомогою веб-камери, що підключена до ПК і безперервно передає в програму послідовність кадрів в реальному часі (25 кадрів/сек);

− локалізація руки - полягає у знаходженні області руки на зображенні захопленому з веб- камери. Звичайні веб-камери не відзначаються якісним зображенням, у зв'язку з чим, спочатку, потрібно обробити зображення, щоб позбутися на ньому завад, які можуть бути викликані зовнішніми умовами, а саме: шуми на зображенні (щоб позбутися шумів здійснюється згладжування зображення) і перепади освітлення (для цього проводиться вирівнювання яскравості зображення). Далі здійснюється перетворення колірних



просторів: RGB простір перетворюється на колірний простір HSV з виділенням каналу S та проводиться нормалізація RGB простору з виділенням каналу R. Для виділення руки на зображенні проводиться сегментація по кольору в каналах RS. Після закінчення сегментації аналізуються отримані об-

ласті, з метою позбутися невірно сегментованих часток зображення і знайти область на зображенні, що відповідає області руки;

− розпізнавання жестів - на отриманому зображенні визначається контур руки і його характеристики, що дозволяє класифікувати жест;

− розпізнаний жест подає відповідну команду програмі.

Локалізація руки на зображенні

Для виділення потрібної області кисті руки було обрано простий в реалізації і виконуваний з високою швидкістю метод сегментації за кольором. З метою вибору найкращої реалізації сегментації були вибрані RGB, HSV і норма- лізований RGB колірні простори.

Розглянемо колірні простори, які використовуються при моделюванні:

а) RGB колірний простір - адитивна колірна модель (кольори виходять шляхом додавання (англ. addition) до чорного), що описує спосіб синтезу кольору для кольоровідтворення. Сегментація шкіри в колірному просторі RGB вважається самою нестійкою і сильно залежить від зовнішніх умов наприклад, освітлення;

б) HSV колірний простір - колірна модель, в якій координатами кольору являються: 1) hue - колірний тон, який варіюється в межах 0 - 360°, проте іноді приводиться до діапазону 0 - 100 (чи 0-1); 2) saturation – насиченість, що варіюється в межах 0 - 100 (чи 0 - 1); 3) value (значення кольору) - яскравість, що задається в межах 0-100 (чи 0-1). Не зважаючи на те, що ця колірна модель дозволяє зменшити вплив освітленості і виділи- ти ділянки шкіри краще, ніж при використанні колірної моделі RGB, вона недостатньо стійка до перепадів освітленості і при роботі з веб- камерою з низьким розрізненням показує не кращі результати.

в) нормалізований RGB - це представлення, яке легко отримати процедурою нормалізації RGB величин.

Нормалізований RGB простір має властивість: rgb + + = 1.

Третя компонента b не містить ніякої важливої інформації і може бути відкинута, тим самим зменшуючи розмірність простору. Компоненти, що залишилися, часто називають "чистими кольорами", оскільки залежність r і g компонент від освітленості зменшена нормалізацією. Особливістю цього представлення є те, що окрім незалежності від навколишнього освітлення, нормалізований RGB незмінний (при певних допущеннях, оскільки неможливо повністю уникнути впливу освітлення) до зміни орієнтації поверхні до джерела світла.

Теоретично, пікселі, які належать до області руки повинні бути близькими по кольору до кольору шкіри і належати певному діапазону на гістограмі.

Для перевірки цього на практиці були захоплені п'ять зображень з камери і на цих зображеннях область руки була виділена вручну. Потім були побудовані гістограми цих зображень. В результаті було встановлено, що пікселі, які належать шкірі, знаходяться в діапазоні яскравості від 80 до 130 в R-каналі, а в G-каналі від 60 до 95. Виявилось, що B-канал не дуже інформативний для сегментації шкіри, тому далі інформація цього каналу не використовувалась. Це допомогло зменшити розмірність і кількість обчислень.

Тестування розроблених функцій проводилось на 3-х вибірках при різних умовах освітлення: - штучне освітлення з розміщенням джерела світла перед об'єктом; - штучне освітлення з розміщенням джерела світла за об'єктом; - в умовах недостатньої освітленості (наприклад, екран монітора).

Об'єднавши результати по всім трьом тестам (табл. 2), та проаналізувавши їх, можна зробити висновок, що сегментація, яка проводиться в R (R канал у нормалізованому RGB просторі) і S (S канал HSV колірного простору) каналах є найбільш стійкою до змін освітлення, а також має найменший відсоток помилок 2-го роду.

Колірний простір	Точність розпізнавання, %	Помилки 1-го роду, %	Помилки 2-го роду, %
RGB	64,03	35,97	23,74
HS	44,87	55,13	20,95
Нормалізований RGB	76,1	23,9	13,87
RS	91,81	8,19	4,03

Таблиця 1. Середній результат сегментації по всім тестам

В результаті етапу моделювання колірний простір RGB виявився самим нестійким до змін освітлення, оскільки в цій моделі представлення компонента яскравості не відокремлена від колірної компоненти. Нормалізований RGB простір частково допомагає уникнути залежності від освітлення. У просторі HSV компонента V віідповідає за яскравість зображення і, якщо проводити аналіз тільки H і S каналу, то результати менше залежатимуть від освітлення, але повністю позбавитися від нього, особливо при роботі з камерою, неможливо. Також, якщо фон містить області співпадаючі з кольором шкіри, вони можуть бути невірно класифіковані. З такими помилками можна впоратися, перевіряючи площу області або її форму. Розпізнавання жестів Для розпізнавання жестів використовується метод розпізнавання з урахуванням характеристик і властивостей контуру. Долоня людини і пальці мають

специфічну форму, тому можна знайти властивості контуру, які будуть однозначно описувати жест.

В результаті аналізу існуючих методів і алгоритмів був розроблений алгоритм, який використовує характеристики контуру (області опуклості та їх дефекти) і співвідношення осей еліпса прямокутника, в який вписана долонь. Для реалізації цього алгоритму необхідно знайти контур долоні, а потім точки опуклостей контуру, які відповідають кінчикам пальців (рис. 11 точки A1, B1, C1, D1, E1). Потім точки опуклостей контуру з’єднуються і утворюють область, яку займає долонь. Далі необхідно обчислити дефекти опуклості (на рис. 12 області A, B, C, D, E, F, G, H). Вони обчислюються як відстань від прямої, що з'єднує дві сусідні точки опуклості, до найглибшої западини (рис. 3 точки A, B, C, D) контуру руки.

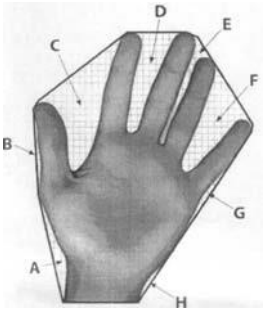


Рис. 11. Фрагмент результату моделювання: контур опуклості (contour conexivity), точки опуклості (Al,Bl,Cl,Dl,El) і точки дефектів опуклості (A, B, C, D)

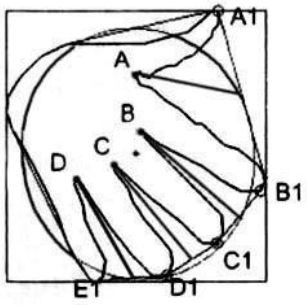


Рис. 12. Контур опуклості долоні (contour conexivity) і області дефектів опуклості (A, B, C, D, E, F, G, H)

За допомогою властивостей опуклостей контуру можна легко вичислити кількість пальців, однак це є недостатньою ознакою для розпізнавання усіх десяти жестів. Тому використовується додаткова ознака: співвідношення осей (більшої осі до меншої) еліпса (або прямокутника), в який вписана долоня.

№	Позначення жесту	Кіль-кість пальців
1	«Start»	0
2	«Stop»	0
J	«Pause»	1
4	«Previous»	1
5	«Next»	2
6	«Volume Up»	1
7	«Volume Down»	2
8	«Close»	4
9	«Forward»	J
10	«Full Screen»	5

Таблиця 2. Набір жестів і їх властивості № Позначення жесту Кількість пальців

У програмної реалізації підсистеми був створений клас GestureRecognition, який дозволяє провести сегментацію зображення у потрібних каналах. Реалізований алгоритм був протестований на 75 зображеннях рук. Результати тестування представленні в табл. 4.

Позначення	Вірно розпізнані, %	Помилки 1-го роду, %	Помилки 2-го роду, %
«Start»	97,33	1,33	1,33
«Stop»	94,66	1,33	4,01
«Pause»	94,66	2,66	2,66
«Previous»	92,0	1,33	6,66
«Next»	94,66	0,0	5,33

Таблиця 3. Результати розпізнавання жестів

Наведено структуру та досліджено роботу системи управління жестами як перспективного інтерфейсу людина-машина, що містить наступні етапи:

1) захоплення зображення;

2) локалізація руки;

3) розпізнавання жестів;

4) формування команд.

Така система дозволяє реалізувати дистанційне управління автоматизованими пристроями не тільки звичайними користувачами, а й людьми з вадами слуху і мови. Обраний для локалізації руки на зображенні метод сегментації за кольором, забезпечує достатню швидкодію для того, щоб його програмна реалізація могла працювати в реальному часі.

Згідно аналізу результатів тестування за різних умов освітлення, яке виконувалось над послідовністю зображень з камери, було визначено, що сегментація, яка проводиться в R (r канал у нормалізованому RGB просторі) і S (S канал HSV колірного простору) каналах є найбільш стійкою до змін освітлення, а також має найменший відсоток помилок 2-го роду. Це свідчить про доцільність застосування саме цього методу сегментації в R і S каналах різних колірних просторів.

Для розпізнавання жестів пропонується алгоритм, який використовує характеристики контуру (області опуклості та їх дефекти) і співвідношення осей еліпса прямокутника, в який вписана долонь. Відповідно до результатів тестування запропонованого алгоритму на 75 зображеннях рук помилки розпізнавання 2-го роду складають не більше 6,5%, що свідчить про надійність та завадостійкість даного методу.

Третій розділ описує сам застосунок, роботу алгоритму та опис засобів реалізації.

Застосунок реалізований на трьох мовах програмування: Swift, Objective-C та C++ з використанням бібліотеки opencv.

Swift — це об'єктно-орієнтована мова програмування, розроблена компанією Apple для того, щоб співіснувати з Objective C і бути стійкішою до помилкового коду. Swift була представлена на конференції розробників WWDC 2014 . Мова побудована з LLVM компілятором, включеного у Xcode 6 beta. Безплатний посібник по мові програмування Swift доступний для завантаження у магазині iBooks.

Компілятор Swift побудований з використанням технологій вільного проекту LLVM. Swift успадковує кращі елементи мов C і Objective-C, тому

синтаксис буде звичний для знайомих з ними розробників, але водночас відрізняється використанням засобів автоматичного розподілу пам'яті і контролю переповнення змінних і масивів, що значно збільшує надійність і безпеку коду.

Objective-C — рефлективна, високорівнева об'єктно-орієнтована мова програмування загального призначення, розроблена у вигляді набору розширень стандартної C.

Розроблена компанією Apple, використовується в основному у Mac OS X та GNUStep — середовищах, розроблених на основі стандарту OpenStep, та Cocoa — бібліотеки компонентів для розробки програм. Програму на Objective-C що не використовує цих бібліотек можна скомпілювати для будь-якої платформи, яку підтримує gcc компілятор з підтримкою Objective-C.

Objective-C є розширенням C і тому будь-яку програму на C можна скомпілювати компілятором Objective-C.

ООП в Objective-C включає інтерфейси, класи, категорії. Реалізовано одиничне, невіртуальне спадкування. Немає єдиного базового класу для всіх об'єктів. Всі методи в класі — віртуальні. Категорія — парадигма яка дозволяє описувати інтерфейс з методами які «необов'язково» імплементувати.

C++ (Сі-плюс-плюс) — мова програмування високого рівня з підтримкою декількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Розроблена Б'ярном Страуструпом (англ. Bjarne Stroustrup) в AT&T Bell Laboratories (Мюррей-Хілл, Нью-Джерсі) у 1979 році та початково отримала назву «Сі з класами». Згодом Страуструп перейменував мову у C++ у 1983 р. Базується на мові C. Визначена стандартом ISO/IEC 14882:2003.

Для використання бібліотеки OpenCV потрібна мова програмування C++. Так як тема моєї роботи це мова програмування Swift, то гра написана на ній, але для того щоб застосунок міг розпізнавати жести руки, потрібно використовувати обидві мови. Swift не має можливості використовувати C++ класи, але може використовувати іншу мову, доступну для розробки під iOS – Objective-C яка має таку можливість.

Тому для використання OpenCV в застосунку одночасно працює код на трьох мовах програмування.

Алгоритм **отримання бінарного** зображення (області що рухається):

1. Згладжуєм поточний кадр (currentFrame) фільтром Гаусса, щоб позбутися від шумів.

2. З згладженого поточного кадру (m_bluredImage) віднімаємо попередній (m_previousImage). Якщо шукані зображення були в градаціях сірого, то і значення пікселів в різниці (mask) будуть змінюватися від нуля до 255 (при восьми бітної глибині кольору).

3. Порівнюємо значення отриманої різниці з деяким граничним значенням. Якщо значення пікселя більше порогового, то цей піксель належить рухомому об'єкту, інакше відкидаємо його. Тепер ми отримали бінарне зображення (mask), де нуль означає, що пиксель не рухається, відмінне від нуля значення - піксель рухається.

4. Застосовуємо морфологічні операції закриття та відкриття, щоб позбутися від рухомих регіонів малого розміру (шуми камери). Отримане зображення (mask) і є рухомий контур.

5. Наносимо бінарне зображення на так зване зображення історії руху (motionHistoryImage). На ньому намальовані рухомі контури за останні, наприклад, 200 мс (m_motionHistoryDuration). Контури були отримані через постійні проміжки часу. Інтенсивність пікселів контуру обернено пропорційна часу, який пройшов від вимірювання контуру до даного моменту. Тобто ніж раніше був отриманий рухомий контур, тим він блідіший зображений на зображення історії руху.

6. Виділяємо регіони (targets) з різними рухами на зображення історії руху.

7. Відкидаємо всі регіони, площа яких менше деякого значення

8. Зберігаємо згладжений поточний кадр як попередній кадр.

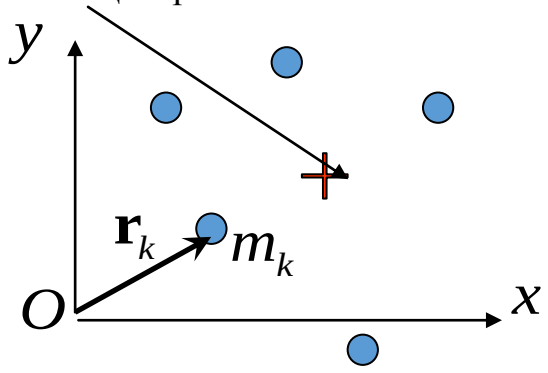
Алгоритм **розпізнавання руху** по бінарному зображенню.

Для розпізнавання жестів був спроєктований та реалізований алгоритм, заснований на підрахунку центру мас.

Це́нтр іне́рції або центр мас системи матеріальних чок масою m_i із радіус-векторами $\mathbf{r}_i$ визначається як

$$\mathbf{r}_c = \frac{\sum_i m_i \mathbf{r}_i}{\sum_i m_i}.$$

Центр мас



На кожному кадрі отримується бінарне зображення рухомої області. Далі воно перетворюється на матрицю в якій кожен єлемент 0 або 1. Далі обчислюється центр мас матриці.

```
int tmpsumx = 0, tmpsumy = 0, allmass = 0;
        for (int i = 0; i < currentFrame.rows; i++)
        {
                for (int j = 0; j < currentFrame.cols; j++)
                {
                        int tempmas = temp_massive_for_matrix[i][j];
                        tmpsumx += tempmas*j;
                        tmpsumy += tempmas*i;
                        allmass += tempmas;
                }
        }
xc = tmpsumx / allmass;
yc = tmpsumy / allmass;
```

xc та yc це координати центру мас.

Після обчислення центру мас отримане значення порівнюється з попереднім і стає срозуміло куди рухається об’єкт на зображенні.

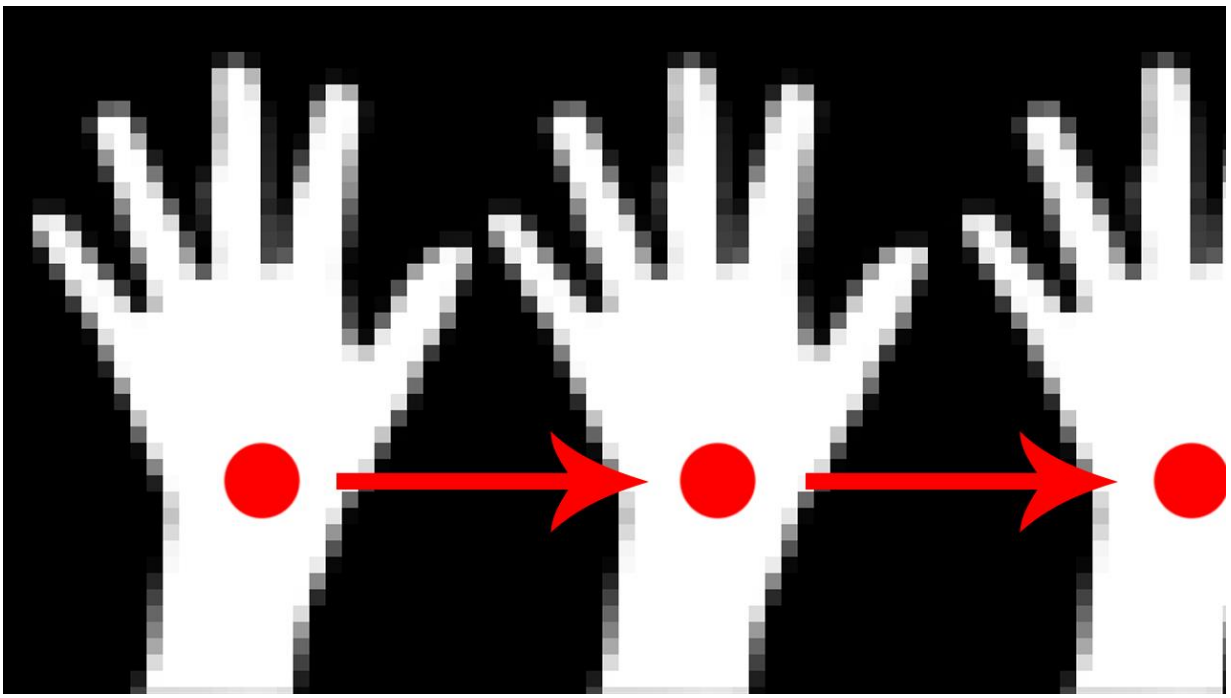


Рис. 13. Зображення руху руки направо.

На даній картинці демонструється рух руки направо. Легко побачити, що центр мас руки теж рухається направо.

Далі визначається напрямок руху центру мас якщо зміни більше ніж 15 пікселів:

```
int delta = 15;
int deltax = xc - oxc, deltay = yc - oyc;

        movex = 0;
        movey = 0;

        if (deltax > delta)
                movex = 1;//std::cout<<" -> ";
        else if (xc - oxc < -delta)
                movex = -1;//std::cout<<" <- ";

        if (deltay > delta)
                movey = -1;//std::cout<<"\\/"<<"\n";
        else if (yc - oyc < -delta)
                movey = 1;//std::cout<<"/\\"<<"\n";
```

Якщо movex > 0 це рух вправо, якщо < 0 то в ліво.

Якщо movey > 0 це рух нагору, якщо < 0 то донизу.

Наступний етап – це пошук середнього значення за останні 25 кадрів.

Наприклад рух руки направо може виглядати так:

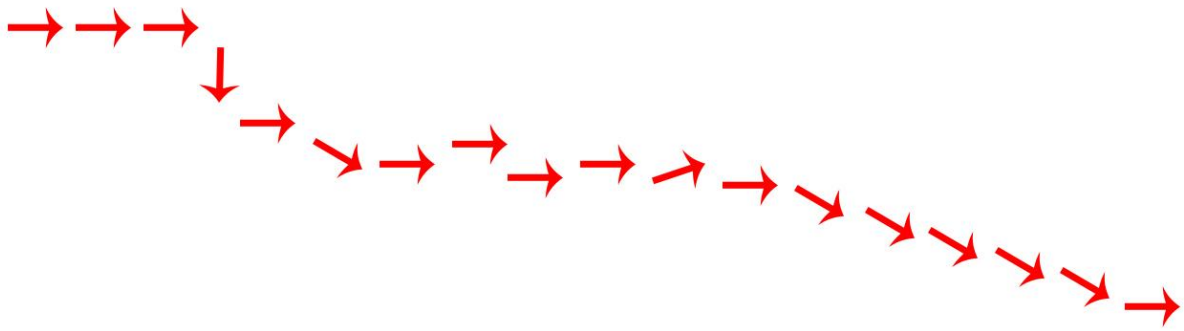


Рис. 14. Можлива траєкторія руху руки.

Тому потрібно знайти середнє значення для того щоб зрозуміти, що це жест направо.

Далі значення зі змінної tempMove оброблюється, перетворюється на команду і передається у гру для демонстрації.

У четвертому розділі розповідається про гру для демонстрації і результати роботи програми.

Для демонстрації можливостей системи розпізнавання жестів потрібна була проста та зрозуміла гра з управлінням чотирма жестами. Такою була вибрана популярна гра «2048».

2048 — браузерна гра, написана 19-річним італійським розробником Габріелем Чіруллі (італ. Gabriele Cirulli) мовою програмування JavaScript. Ігрове поле має форму квадрата 4x4. метою гри є отримання плитки номіналу «2048» (але при бажанні можна продовжити далі). Код гри відкритий і викладений на сторінці розробника в GitHub. Гра «2048» була написана менш ніж за два дні в якості вправи в програмуванні за словами автора, наведеним газетою Los Angeles Times, Чіруллі порахував своє творіння «випадковим вторгненням в ігрову індустрію»і не планує надалі займатися розробкою ігор.

Прообразом «2048» є комерційна гра Threes її творці залишилися незадоволеними успіхом «2048» і назвали гру Чіруллі «зіпсованим плагіатом». Після виходу Threes в App Store з'явилося кілька її клонів, в тому числі ігри «1024» і «2048», створені іншими розробниками при розробці своєї версії Чіруллі надихався цими іграми

Правила гри

- На кожному рівні з'являється плитка номіналу «2» (з ймовірністю 90%) або «4» (з ймовірністю 10%)
- Натисканням стрілки гравець може скинути всі плитки ігрового поля в одну з 4 сторін. Якщо при скиданні 2 плитки одного номіналу "налітають" одна на іншу, то вони зливаються в одну, номінал якої дорівнює сумі злитих. Якщо при натисканні кнопки, місце розташування плиток і їх номінал НЕ зміниться, то хід не звершується
- За кожне злиття ігрові очки збільшуються на номінал новоутвореної плитки
- Гра закінчується, якщо після чергового раунду неможливо вчинити дію.

Гра буда реалізована на мові програмування Swift.

Скріншот ігрового поля:

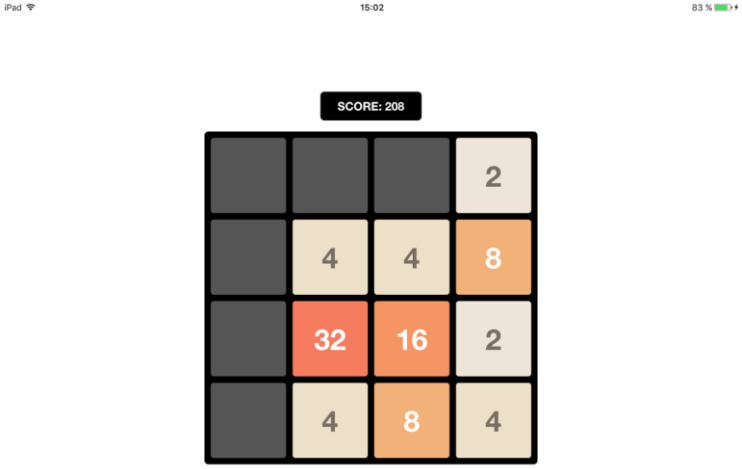


Рис. 15. Скріншот застосунку.

Результати дослідження

Розроблений застосунок з високою точністю розпізнає жести рук перед пристроєм за допомогою фронтальної камери. Розроблений алгоритм можли-

во застосувати у будь-якому застосунку на будь-якій платформі. Від ігор до утиліт і управління операційною системою.

Досліджені можливості засобів розробки під платформу iOS, такі мови програмування як Swift, Objective-C та C++.

Очевидно, що впровадження Swift помітно позначиться на всій програмній екосистемі Apple. Оскільки розробка додатків для iOS і OS X стає все простіше і легше, багато професіоналів в інших мовах і платформах будуть мати бажання спробувати свої сили на новому полі діяльності.

Це означає, що до мобільних і комп'ютерних платформ Apple буде залучено більше число розробників. Більше розробників - це більше додатків і більше вибору для споживача. А от питання про якісний рівень такого софта доведеться залишити відкритим. Поки що Swift занадто нова технологія і вона ще не освоєна навіть професійними програмістами для пристроїв Apple. До того ж вона більш доступна в розумінні для студентів та інших людей, які роблять перші кроки в програмуванні. До чого все це може призвести, поки сказати важко.

Досліджені можливості IDE Xcode. Це одна з найкращих IDE яка на равних конкурує, а в деяких моментах і випереджає свої аналоги.

Реалізовано дві принципово різні гри абсолютно різними методами і підходами.

Досліджено роботу систем управління жестами як перспективного інтерфейсу людина-машина, що містить наступні етапи: 1) захоплення зображення; 2) локалізація руки; 3) розпізнавання жестів; 4) формування команд. Такі системи дозволяють реалізувати дистанційне управління автоматизованими пристроями не тільки звичайними користувачами, а й людьми з вадами слуху і мови.

Розроблений алгоритм розпізнавання жестів оснований на зміненні зображення(виділяються тільки рухомі частини зображення). Алгоритм дозволяє чітко розпізнавати як швидкі так і повільні жести та може бути застосований у реальному житті, наприклад для керування грою 2048 як представлено у даній роботі.

Комп’ютерний зір це дуже перспективний напрямок розвитку сучасних технологій і може використовуватися у будь-якій сфері діяльності від звичайних ігор та утиліт до управління складною операційною системою.

Заключення і висновки.

1. Досліджені можливості засобів розробки під платформу iOS, такі мови програмування як Swift, Objective-C та C++ і IDE Xcode.
2. Визначені переваги Swift перед Objective-C.
3. Досліджені інструменти для обробки зображення. Інтегровано бібліотеку OpenCV в проект під iOS.
4. Досліджено перспективи розвитку мови програмування Swift та її вплив на індустрію.
5. Досліджено роботу систем управління жестами як перспективного інтерфейсу людина-машина.
6. Розроблений авторський алгоритм розпізнавання жестів.
7. Реалізований за стосунок в якому на прикладі гри 2048 реалізоване управління жестами через фронтальну камеру пристрою.

РЕФЕРАТ

Сторінок: 103
Рисунків: 28
Таблиць: 6
Використаних джерел: 26

Мета роботи: Дослідити можливості засобів розробки мобільних застосунків. Дослідити роботу систем управління жестами та запропонувати авторський алгоритм вирішення даної задачі. Розробити застосунок, який розпізнає рух об’єктів перед фронтальною камерою мобільного пристрою за допомогою бібліотеки OpenCV під операційну систему iOS.

Результати: Розроблений застосунок, який з високою точністю розпізнає жести рук перед пристроєм за допомогою фронтальної камери. Розроблений алгоритм можливо застосувати у будь-якому застосунку на будь-якій платформі. Від ігор до утиліт і управління операційною системою.

Досліджені можливості засобів розробки під платформу iOS, такі мови програмування як Swift, Objective-C та C++. Реалізовано дві принципово різні гри абсолютно різними методами і підходами.

Досліджено роботу систем управління жестами як перспективного напрямку розвитку інтерфейсів управління.

Публікація: Романенко Ю.О., магістрант гр. СП-14м, Попівщий В.І., доц. - науковий керівник. «Вплив мови програмування SWIFT на IT-індустрію» (ст. 39). Матеріали XX науково-технічної конференції студентів, магістрантів, аспірантів і викладачів ЗДІА «ЕЛЕКТРОНІКА, АВТОМАТИЗОВАНІ СИСТЕМИ ТА СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ» том III, Запоріжжя, ЗДІА, 2015.

РЕФЕРАТ

Страниц: 103
Рисунков: 28
Таблиц: 6
Использованных источников: 26

Цель работы: Исследовать возможности средств разработки мобильных приложений. Исследовать работу систем управления жестами и предложить авторский алгоритм решения данной задачи. Разработать приложение, которое распознает движение объектов перед фронтальной камерой мобильного устройства с помощью библиотеки OpenCV под операционную систему iOS.

Результаты: Разработанный приложение, с высокой точностью распознает жесты рук перед устройством с помощью фронтальной камеры. Разработанный алгоритм можно применить в любом приложении на любой платформе. От игр до утилит и управления операционной системой. Исследованы возможности средств разработки под платформу iOS, такие языки программирования как Swift, Objective-C и C ++. Реализованы две принципиально разные игры совершенно разными методами и подходами. Исследована работа систем управления жестами как перспективного направления развития интерфейсов управления.

Публикация: Романенко Ю.А., магистрант гр. СП-14м, Попивщий В.И., доц. - Научный руководитель. «Влияние языка программирования SWIFT на ИТ-индустрию» (ст. 39). Материалы XX научно-технической конференции студентов, магистрантов, аспирантов и преподавателей ЗГИА «ЭЛЕКТРОНИКА, АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ И СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ» том III, Запорожье, ЗГИА, 2015.

THE ABSTRACT

Pages: 103
Figures: 28
The tables: 6
The references: 26

The purpose of work: To explore the potential of developing mobile applications. Investigate the job of management to offer gestures and author algorithm for solving this problem. Develop application that detects motion of objects in front of the front camera mobile device using OpenCV library under the operating system iOS.

Results: The developed application that accurately recognizes hand gestures to the device using the front camera. The algorithm may be applied in any application on any platform. From games to utilities and operating system.
The possibilities of development tools under the platform iOS, programming languages such as Swift, Objective-C and C ++. Implemented two fundamentally different game entirely different methods and approaches.
Studied the work management systems as a promising gesture towards the development of management interfaces.

Publication: Romanenko Y., SP-14m, Popivschyy VI, Assoc. - supervisor. "The impact of programming languages SWIFT IT industry" (p. 39). Materials XX scientific conference of students, undergraduates, graduate students and faculty DIG "Electronics, automated systems and modern information technology" Volume III, Zaporozhye, DIG, 2015.