УДК 621.38(62-52)

С.А. Нечипорчук, Т.А. Ліхоузова

УПРАВЛІННЯ КУРСОРОМ ЗА ДОПОМОГОЮ ЗОРОВОГО ТРАКТУ

Анотація : Розглядається метод керування курсором за допомогою зорового тракту з використанням веб-камери.

Ключові слова: комп’ютерне бачення, зоровий тракт, очі, керування курсором.

Вступ

Сьогоднішні комп’ютерні інтерфейси в переважній більшості являються контактними, тобто такими, для яких необхідний безпосередній контакт за допомогою миші, сенсорних екранів, трекпадів, тощо. Саме тому вони являються дуже повільними та не природними. Але з темпом розвитку технологій постає необхідність в більш натуральних способах взаємодії людини і комп’ютера.

Алгоритм та програмний комплекс, описаний в даній роботі, дозволяє керувати курсором за допомогою погляду. Він аналізує потік кадрів веб-камери, направленої на користувача, аналізує положення очей та переміщує курсор в розраховане положення. Даний тип інтерфейсів може бути використаним як для людей з обмеженими властивостями, так і як інтерфейсів вищого рівня.

Аналіз існуючих рішень

На сьогодні існує лише один програмний комплекс, який дозволяє керувати курсором за допомогою погляду: “Cameramouse”, але цей комплекс має ряд недоліків.

В основі даного комплексу лежить алгоритм пошуку ключових точок та опису їх дескрипторів, завдяки якому дана програма може відслідковувати рухи не тільки ока, а й рухи будь-яких інших елементів кадру. Це накладає важливе обмеження на роботу всієї програми: перед початком користування необхідно вказати на зображенні з камери, що саме потрібно відслідковувати. Це необхідно зробити якимось іншим маніпулятором (наприклад, мишею). Але люди з обмеженими можливостями не мають змоги виконати такі дії.

Дане програмне забезпечення не є кросплатформенним. Тобто користувачі з іншими операційними системами не зможуть ним користуватися.

Відсутність відкритих вихідних кодів. Не дивлячись на попередній недолік, при наявності відкритих програмних кодів можливо було перекомпілювати їх на новій платформі.

Іншим недоліком є складність та ресурсоємність самого алгоритму. Алгоритм полягає в знаходженні ключових точок та їх дескрипторів

 ISSN 1560-8956

на фрагменті зображення, який буде відслідковуватися. Потім знаходяться ключові точки та дескриптори кожного кадру та порівняння їх з дескрипторами фрагменту. Після чого знаходиться область де їх різниця найменша.

Для компенсації перерахованих недоліків було прийнято рішення розробити новий програмний комплекс, оскільки модифікація “Cameramouse” неможлива.

Постановка задачі

Об’єктом управління є курсор. Алгоритм управління повинен знаходити область погляду користувача через веб-камеру та переміщувати курсор в область погляду. Він повинен мати такі характеристики:

- точність знаходження очей на зображенні повинна дорівнювати 1 пікселю. Така точність зумовлена тим, що в середньому масштаб між рухами очей та рухами курсору дорівнює 1:9. Тобто, якщо точність знаходження очей на зображенні буде дорівнювати 2 пікселям, то похибка переміщення курсору буде дорівнювати 9 пікселів. Така похибка є неприпустимою тому, що користувач не зможе нормально перемістити курсор на елементи розмір яких близький до 9 пікслеів та менше;
- частота роботи алгоритму повинна бути не меншою ніж 25 Гц. Це необхідно тому, що даний алгоритм буде використовуватися в системі реального часу. Дана система буде зв’язана з веб-камерою. Камера повинна мати частота 25 Гц. Якщо ж частота алгоритму буде менше, то переміщення курсору буде виконуватися з постійно зростаючою затримкою;
- компенсація випадкових рухів голови. Дана вимога зумовлена тим, що користувач не може нерухомо тримати голову (наприклад, хоча б, за рахунок дихання). Якщо такі рухи не компенсувати, то курсор комп’ютера буде випадково переміщатися на невеликі відстані. Дані “шуми” будуть заважати нормальній роботі користувача;
- знаходження області очей на зображенні з будь-яким адекватним фоном. Адекватним фоном будемо називати будь-який фон, на якому не зображено чітких очей, голови людини. Дана вимога є важливою тому, що дозволить використовувати даний алгоритм майже з будь-яким оточенням;
- адаптація до різних рівнів освітлення;
- оброблення невеликих поворотів голови. При роботі з комп’ютером користувачі часто виконують деякі повороти головою, наприклад, при переміщенні погляду з лівого краю до правого. Необхідно коректно обробляти такі повороти так, щоб користувач не відчував незручностей.

Програмне забезпечення, що реалізує алгоритм, повинно мати наступні характеристики:

- кросплатформенність. Дане програмне забезпечення повинно працювати на будь-якій платформі (Linux, Mac, Windows) з архітектурою x86 або AMD64. Це дозволить користувачам популярних платформ користуватися даним ПЗ;
- зручність у користуванні. Інтерфейс користувача повинен бути мінімальним та зручним. Такий інтерфейс дозволить людям, які не мають великого досвіду роботи з комп’ютером користуватися даним програмним забезпеченням;
- зручність у встановлення. Необхідно врахувати те, що пересічний користувач не зможе сам повністю зкомпілювати всі залежності, потрібні для роботи. Тому необхідно зробити процес компіляції та встановлення якомога простішим;
- робота з будь-якими веб-камерами. Це дозволить користувачам використовувати веб-камери будь-якого виробника та будь-якого типу.

Опис алгоритму

Базовою задачею при реалізації програмного комплексу є пошуку очей в відеопотоці. Існують різні методи пошуку очей на кадрі в відеопотоці, розглянемо найбільш загальні серед них. При виборі методів акцент зробимо на швидкість роботи, адаптивність до рівня освітлення, спроможності коректування хибних рухів. Отже, всі методи можна розділити на 3 групи:

- методи з використання шаблонів (template based methods);
- методи, основані на пошуку оточення області очей (appearance based methods);
- методи, основані на пошуку особливостей області очей (feature based methods).

В методах з використанням шаблонів спочатку формується модель очей, основана на їх формі. Далі на зображенні шукається зона, яка є найбільш схожа на початковий шаблон по деякому критерію. Зазвичай даним критерієм є сума модулів різниць. Якщо сума менша за деякий поріг, то дана область на зображенні вважається шуканою. Метод темплейтів є надшвидким, простим методом, але його точність менша ніж точність каскадів Хаара. Він не є інваріантним відносно поворотів в будь-якій площині.

Методи, основані на пошуку оточення області очей, використовують фотометричні властивості шуканих фрагментів. Зазвичай дані методи потребують великої кількості тренувальних даних. Для пошуку очей це можуть бути зображення очей різних людей, при різних орієнтаціях та освітленні. Ці дані використовуються для тренування класифікатора (наприклад, нейронної сітки). Найбільш вдалий представник цієї групи методів – метод каскадів Хаара. Недоліком метода каскадів Хаара є його невелика точність. В силу цього, на кожному кадрі знайдена область

 ISSN 1560-8956

випадково переміщується в діапазоні 1–5 пікселів. Для умов даної задачі така похибка є недопустимою.

Методи, основані на пошуку особливостей області очей, знаходять ці характеристики (границя та контрастність зіниці, розподіл освітлення білка, та інше) для визначення вектора властивостей зображення з очима. Такі методи являються дуже ефективними, хоча й потребують більше процесорного часу ніж інші. Найбільш поширений представник цієї групи методів – метод SURF, який має дуже високу точність. Також він дозволяє знаходити фрагменти на кадрі які повертаються не тільки в площині зображення, але й в будь-якій іншій площині. Недоліком є невисока швидкодія, навіть з застосуванням інтегральних зображень кожен кадр довго оброблюється. Це не задовольняє умови поставленої задачі.

В силу недоліків перерахованих методів було прийнято рішення використати їх комбінацію: спочатку знаходиться початковий шаблон методом каскадів Хаара. Далі методом темплейтів знаходиться наступний шаблон, який буде використовуватися в наступній ітерації як початковий. Кожні 50 кадрів даний шаблон коректується методом каскадів Хаара. Перевагами такого комбінованого методу є:

- висока швидкодія, завдяки використанню метода темплейтів;
- висока точність, завдяки використанню динамічних темплейтів та коректування їх за допомогою методу каскадів Хаара;
- інваріантність відносно поворотів голови в будь-яких площинах, отримана завдяки невеликому дрейфу шаблона.

До недоліків можна віднести невелику швидкість реакції системи на зміну положення голови. Тобто, при різких рухах голови даний метод не точно знайде шуканий фрагмент на кадрі, або зовсім не знайде його. Це викликано тим, що динамічний метод темплейтів може знаходити нові шаблони, які мають малу відмінність від поточного. При різкій зміні області очей даний фрагмент може бути не знайдений.

Для реалізації методу був розроблений власний алгоритм, зображений на рис. 1. Основні моменти даного алгоритму описані нижче.

Конвертування кадру з RGB-простору до сірого необхідно для того, щоб зменшити навантаження на процесор. RGB-простір має три канали, для кожного з яких необхідно було б проводити нормалізацію, пошук каскадами Хаара, тощо. А у сірого простору є лише один канал, що значно збільшує швидкодію даного алгоритму. Важливо зазначити, що при переході від трьох-канального простору до одно-канального втрачається певна інформація, але в даному алгоритмі така втрата є допустимою.

На наступному кроці необхідно вирівняти освітлення на всьому зображенні. Це виконується за рахунок нормалізації. За допомогою неї на зображенні згладжуються різкі перепади в освітленні. Так, наприклад, якщо одна сторона обличчя користувача освітлена більше ніж інша, то нормалізація змінить освітлення таким чином, щоб воно було рівним на обох половинах обличчя.

Для знаходження першого шаблону використовуються каскади Хаара. Даний метод при нормальних умовах гарантовано знайде область з очима. Якщо ж з якоїсь причини область не була знайдена, то даний алгоритм повторюватиме спробу до тих пір, поки не буде знайдений перший фрагмент. В кожній наступній ітерації не буде шукатися новий шаблон, а буде використовуватися знайдений на попередній ітерації.

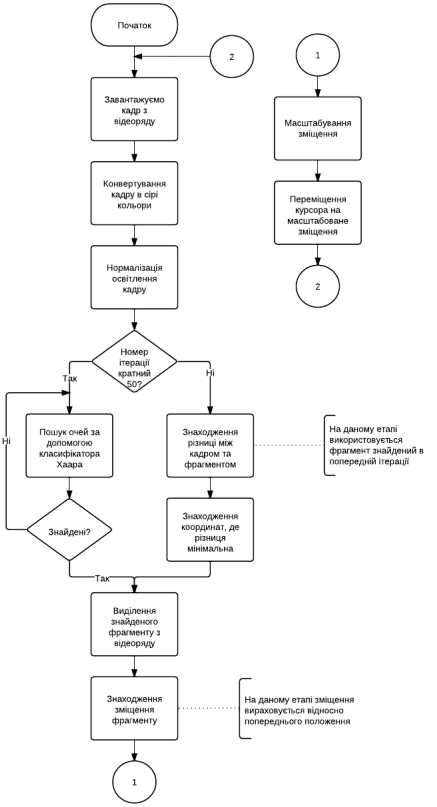

Рис. 1 – Блок-схема роботи алгоритму.

В наступних ітераціях для знаходження фрагменту з очима буде використовуватися метод темплейтів. Від зображення кадру буде відніматися шаблон області з очима, знайдений на попередній ітерації. Область, де дана різниця буде мінімальною і є шукана область. В наступній ітерації вже ця (знайдена) область буде використовуватися як шаблон. Тобто, в даному випадку алгоритм динамічно оновлює шаблон. Такий хід до-

зволяє дуже плавно і точно відслідковувати рухи області з очима. Але з часом даний шаблон буде зміщуватися відносно області з очима. Це викликано тим, що різниця між початковим шаблоном та поточним з кожною ітерацією збільшується. Тобто, даний шаблон дрейфує і може зміщуватися на доволі великі відстані. Щоб уникнути цього, в даному алгоритмі реалізоване оновлення шаблону за допомогою каскадів Хаара кожні 50 кадрів. Дана кількість кадрів була вибрана експериментально. При такій кількості кадрів дрейф шаблону є мінімальним.

На наступному кроці необхідно вирахувати, на яку відстань змістився погляд відносно попередньої ітерації. Для цього потрібно запам’ятати координати погляду попередньої ітерації, після чого знаходити відстань від них до поточного положення. Потім масштабувати дане зміщення для переводу рухів погляду в рухи курсору. В даному алгоритмі зміщення по осі масштабується з коефіцієнтом (-8) Знак мінус присутній тому, що камера отримує зображення, яке є дзеркальним відображенням дійсного. Зміщення по осі Oy масштабується з коефіцієнтом 8. Дані значення були вибрані в ході експериментів. Були враховані найбільш поширені розміри моніторів (1024x768, 1280x800, 1280x1024) та середні розміри обличчя користувачів. При таких коефіцієнтах рух курсору є плавним та достатньо точним.

Для реалізації алгоритму розроблено програмний комплекс на базі ядра Linux, тому жорстких вимог щодо апаратного забезпечення немає. Програмне забезпечення створено на мові програмування C++. Воно задовольняє поставлені вимоги, а саме: кросплатформеність; зручність у використанні (це досягнуто за рахунок повної відсутності інтерфейсу, тобто, щоб використовувати дане програмне забезпечення необхідно лише його запустити); простота у встановленні; робота з будь-якими вебкамерами.

Результати роботи програми

Для оцінки ефективності розробленого програмного комплексу виконано його порівняння з вже існуючим рішення – програмним комплексом “Cameramouse”. Всі тести проводилися на одному комп’ютері з процесором в 1.3 ГГц, з пам’яттю в 2 Гб. Використовувалася веб-камера “Logitech C270”. Дана камера має максимальне розширення зображення 1280*800, а також апаратну реалізацію нормалізації зображення.

В першому тесті було переглянуто навантаження на пам’ять та процесор при нормальному режимі роботи. На рис. 2 зображена гістограма отриманих значень.

Видно, що алгоритм, розроблений в даній роботі, є менш ресурсоємним. Це зумовлено тим, що був вибраний модифікований метод темплейтів для пошуку області з очима. З гістограми видно, що розроблений алгоритм використовує менше пам’яті ніж алгоритм програмного комплексу “Cameramouse”.

На рис. 3. зображена похибка в знаходження області очей зі зміною освітлення.

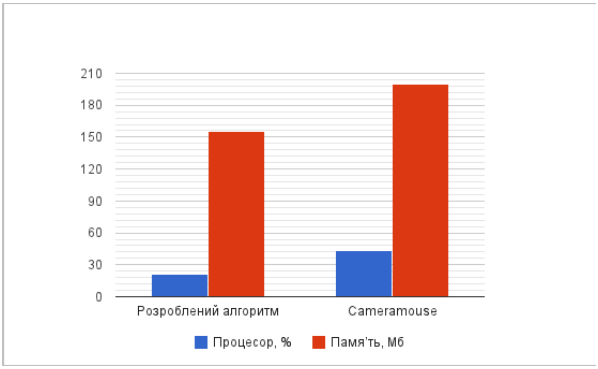

Рис. 2 – Гістограма ресурсів, які використовують розглянуті алгоритми

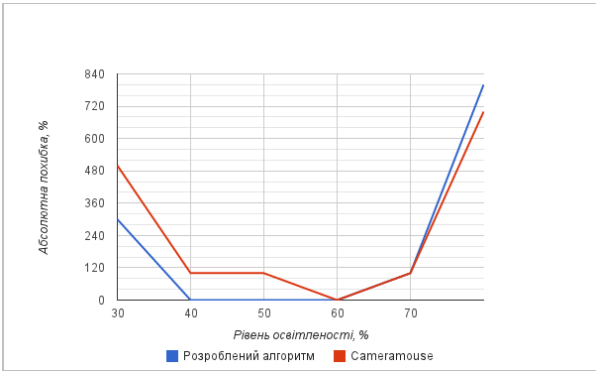

Рис. 3 – Абсолютна похибка відносно рівня освітленості

На даному графіку використовувався рівень освітленості зображення в сірому просторі. Тобто, 0% освітленості має абсолютно чорне зображення, а 100% – абсолютно біле. З графіка видно, що при нормальному рівні освітлення алгоритм, розроблений в даній роботі, має меншу похибку ніж алгоритм, використаний в програмному комплексі “Cameramouse”. Така похибка дозволяє керувати курсором при доволі широкому діапазоні освітлення.

Висновки

Розроблено програмний комплекс для відслідковування погляду користувача та переміщення курсору в зону погляду. В основі лежить метод, отриманий інтегруванням методу каскадів Хаара та модифікованого методу темплейтів. Даний метод дозволяє швидко і точно знайти положення очей та перемістити курсор. Також, на відміну від методу,

 ISSN 1560-8956

використаного в ПЗ “Cameramouse”, даний метод автоматично знаходить область з очима.

Для покращення алгоритму можливо реалізувати “пам’ять”, тобто запам’ятовувати певні ключові знайдені шаблони, шукати кожен з них на зображенні та вибирати кращий за деяким критерієм. Це дозволить виключити з алгоритму метод Хаара для уточнення положення області очей.

Для покращення програмного забезпечення можливо реалізувати імітацію натискання на кнопки миші. Це можна зробити кількома способами, наприклад, виконувати клік миші тоді, коли користувач не рухає курсор на протязі 2-3 секунд, або реалізувати голосове керування, тобто виконання таких команд як “Натиснути”, “Скопіювати”, та інші. Реалізація одного з цих способів дозволить людям з обмеженими можливостями не тільки переміщати курсор, а й виконувати інші операції, які може виконувати миша або інший маніпулятор.

Література

1. Вапник В. Н., Червоненкис А. Я. Теория распознавания образов. — М.: Наука, 1974. — 416 с.
2. Форсайт Дэвид А., Понс Джин. Компьютерное зрение. Современный подход - Computer Vision: A Modern Approach. — М.: Вильямс, 2004. — 928 с.
3. Бабак В.П.,Ханадецький В.С.,Шрюфер Е. Обробка сигналів. К.: Либідь,1996.-392с.
4. Анисимов Б.В., Курганов В.Д., Злобин В.К. Распознавание и цифровая обработка изображений. М: Высшая школа, 1983. –295с.
5. Прэтт У. Цифровая обработка изображений. М.: Мир,1982. – Кн.1,312с.

Отримано 10.04.2012 р.