

ЗАСТОСУВАННЯ СИСТЕМИ ВІЗУАЛЬНОГО МОДЕЛЮВАННЯ У МЕДИЦИНІ

* Київський національний університет імені Тараса Шевченка, Київ, Україна

Анотація. У статті запропонована система візуального моделювання для проведення медичних досліджень у реальному часі. Зокрема, розроблено бібліотеку алгоритмів обробки зображень для застосування в мануальній терапії.

Ключові слова: відеозображення, відеокамера, відхилення контуру, контурні точки, черепні кістки, мануальна терапія.

Аннотация. В статье предложена система визуального моделирования для проведения медицинских исследований в реальном времени. В частности, разработана библиотека алгоритмов обработки изображений для применения в мануальной терапии.

Ключевые слова: видеоизображение, видеокамера, отклонение контура, контурные точки, черепные кости, мануальная терапия.

Abstract. In this article we propose a visual modeling system for real-time medical research. In particular, the library of picture processing algorithms for applying in manual therapy was developed.

Keywords: video picture, video camera, contour deflection, contour points, cranial bones, manual therapy.

1. Вступ

Постановка проблеми. На сьогодні проблема використання сучасних інформаційних технологій у різноманітних галузях практичного напрямку є актуальною. Зважаючи на стрімкий розвиток інформаційних технологій, програмного забезпечення та потужностей сучасної комп'ютерної техніки, використання передових інформаційних технологій являється якісним фактором рівня розвитку більшості галузей практичного напрямку і особливо медицини.

Аналіз останніх досліджень. Сучасні технічні можливості дозволяють медичним технологіям вийти на абсолютно новий рівень. Перш за все, до таких технологій можна віднести діагностику, аналіз, прогнозування розвитку хвороб тощо. Деякі передові інформаційні дослідження уже почали знаходити своє місце в медицині, наприклад, використання нейронних мереж та можливостей штучного інтелекту. До таких досліджень можна віднести розвиток експертних систем [1]. Ці системи дозволяють вирішувати задачі діагностики, диференціальної діагностики, прогнозування, вибору стратегії і тактики лікування та ін. Тут важливим моментом є методи та алгоритми по обробці різноманітних медичних зображень. На цю тему було проведено деякі дослідження, які в основному зосереджувались саме на використанні нейронних мереж для обробки різноманітних статичних медичних зображень. У даній області основну увагу звертають на визначення оптимальних параметрів при навчанні нейронних мереж. Об'єктами досліджень являються зображення легенів. Алгоритмічною основою програмного забезпечення таких систем є використання дискретного перетворення Фур'є [2].

У даній роботі увага акцентується на зовсім інших аспектах обробки зображень у медичних технологіях. А саме: використання сучасних інформаційних технологій для обробки відеозображення, візуального та графічного моделювання в різних областях практичної медицини. На відміну від існуючих підходів, які базуються на статичному використанні зображень (як то томографія), в запропонованому підході досліджується обробка динамічних відеозображень з подальшим застосуванням у медичній практиці. Наприклад,

такий підхід можна використати у хірургії, для зручності та безпеки проведення складних операцій, зокрема, це стосується операцій на людському черепі. Для цього хірургу потрібна детальна інформація про вражені частини, загальна і точна характеристика голови пацієнта та інша додаткова інформація. Сучасні технології дозволяють створювати програмне забезпечення, яке зможе зробити це швидко та ефективно, використовуючи лише ресурси звичайних персональних комп'ютерів. Навіть при проведенні складних операцій при використанні такого програмного забезпечення хірург у будь-який момент часу може уточнити зміну потрібних йому параметрів та характеристик. Це всього лиш один з прикладів застосування вищеописаного підходу. Використовуючи сучасні технічні можливості, ми могли б значно підвищити ефективність, швидкість та зручність проведення аналізу досліджуваних процесів у медицині.

Метою статті є розробка алгоритмічних інструментів (програмне забезпечення) та моделі досліджень для застосування в мануальній терапії.

Новизна та ідея. Основна ідея запропонованого підходу полягає у розробці відповідних алгоритмів та програмного забезпечення саме в області візуального моделювання для подальшого дослідження та практичного використання їх у медицині, наприклад, у хірургії та мануальній терапії. На відміну від попередніх досліджень в цій області, використання даного програмного забезпечення може проводитись безперервно в реальному часі. Наприклад, прямо під час операції хірург зможе отримати необхідні йому результати в будь-який момент часу. Окрім того, запропоноване програмне забезпечення має гнучкі засоби для налаштування до відповідної специфіки задачі, що робить експлуатацію інструментальних засобів простими і доступними. Для використання вже готового програмного забезпечення достатньо лише потужності середнього персонального комп'ютера. При цьому дані алгоритми відразу готові до застосування, без попереднього їх навчання з використанням нейронних мереж та інших дій з боку користувача. Таким чином, на відміну від попередніх досліджень, дане програмне забезпечення має вирішувати великий клас задач та водночас бути максимально простим у застосуванні для досвідченого користувача.

2. Формалізація постановки задачі

Формальна постановка задачі полягає в наступному.

Постановка задачі. На вході маємо зображення контуру голови пацієнта, знятого відеокамерами за допомогою рівномірно розташованих світлодіодів. Кожен світлодіод відповідає певній черепній кістці пацієнта. Лікарю необхідна інформація про те, яка саме з кісток має відхилення від стандартного положення, а також довжину, напрямок та кут відхилення цієї частини черепа. Окрім того, необхідно мати інформацію про довжину верхнього, нижнього та бокових контурів голови, а також довжину та різниці півосей відносно центра та загальну довжину всього контуру.

Необхідно розробити набір алгоритмів, які б у повному обсязі розв'язували сформульовану задачу.

3. Метод розв'язання задачі

Спочатку необхідно розробити загальний алгоритм, який би дозволяв знаходити та аналізувати відхилення. Алгоритм буде виконуватися для кожного кадру нашого відеозображення. Всю роботу алгоритму можна умовно поділити на 3 етапи: розпізнавання та знаходження контурних точок на зображенні; аналіз коректності розміщення контурних точок; визначення відхилень.

Означення. Контурними точками умовно назвемо ті невеликі області зображення, що відповідають світлодіодам.

Перш за все задача зводиться до того, щоб виділити необхідну множину контурних точок для їх подальшого аналізу. Вхідними даними будуть відповідні діапазони кольору в форматі RGB, які відповідають шуканим точкам.

3.1. Алгоритм визначення контурних точок

1. У першу чергу знаходимо центральну точку, яка позначатиме центр нашої вісі координат, а потім вже всі інші точки.

2. Пошук будь-якої точки можна розділити на два пункти: виділення контурів (потрібні нам кольори на зображенні), їх аналіз і знаходження серед них наших точок.

2.1. Виділення контурів здійснюється достатньо просто. Використовуючи копію нашого зображення, ми перевіряємо кожен із пікселів, чи входить він у той кольоровий діапазон, що відповідає нашим точкам. Якщо входить, замальовуємо його білим кольором, інакше – чорним. Таким чином, ми отримуємо зображення, яке містить усі потрібні нам контури.

2.2. Аналізуємо кожен із виділених контурів, щоб виявити контурні точки. Спочатку просто аналізуємо периметр (загальну довжину) нашого контуру, щоб визначити, чи підходить ця точка нам за розміром. Якщо так, тоді залишилось лише знайти центр нашої точки. Для цього використовуємо таке поняття, як моменти.

Означення [3]. Момент – це сумарна характеристика за всіма пікселями контуру. Момент (p, q) визначається як

$$m_{p,q} = \sum_{i=1}^n I(x, y) x^p y^q ,$$

де p – порядок x , q – порядок y , де порядок означає потужність, на якій відповідний компонент взятий у сумі з іншими відображеннями. Використовуючи моменти, дуже просто знайти центр мас будь-якого контуру. Координати центра точки обчислюватимуться так:

$$x = m_{1,0} / m_{0,0},$$

$$y = m_{0,1} / m_{0,0}.$$

Таким чином, у результаті обробки зображення на даному етапі ми отримали певний набір точок $A_1 \dots A_n$. Окрім цих точок, ми отримуємо ще одну точку C , яка означатиме центр координат.

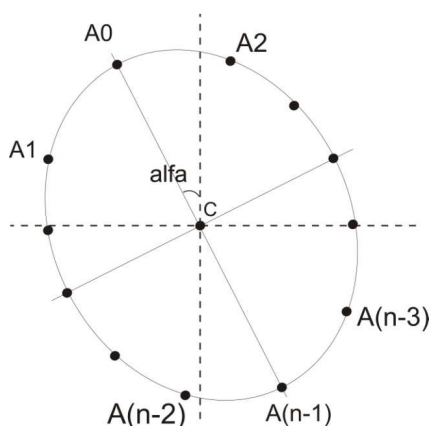


Рис. 1. Відхилення положення системи координат

3.2. Аналіз коректності розміщення контурних точок

Зауважимо, що для чіткої роботи загального алгоритму кількість знайдених точок має бути парною (адже в основному положенні всі точки мають лежати симетрично). Якщо це так, продовжимо виконання наступних етапів алгоритму, інакше ми отримуємо недостатньо чітке або помилкове зображення. Для цього необхідно зробити корекцію, врахувавши відхилення.

Оскільки ми маємо справу з динамічним зображенням, наші точки можуть мати спотворені координати. Наприклад, пацієнт може повернути голову тощо. Тобто система координат швидше за все буде нахилена в той чи інший бік (рис. 1). Для коректної роботи програми ми повинні перейти до нової системи координат,

в якій і будемо проводити усі обчислення.

Алгоритм корекції

1. Нам відомі координати центра C . Для знаходження кута відхилення системи координат необхідно щонайменше знайти верхню вершину овалу (рис. 1). Виходячи із специфіки нашої задачі (наше зображення – це череп), обидві вершини будуть знаходитись на максимальній відстані від центра.

2. Щоб знайти координати верхньої вершини, проходимо по точках, які знаходяться вище або на одному рівні з центром C , обчислюючи відстань між ними і C . Максимальна відстань відповідатиме нашій вершині.

3. Обчислюємо кут α – відхилення центра C відносно нашої системи координат. Нехай $\varepsilon = 2^\circ$ – наша похибка. Якщо $\alpha > \varepsilon$, переходимо до нової системи координат, в якій і будемо шукати необхідні відхилення. Інакше відхилення незначне, і система координат залишається старою.

4. Визначаємо сторону, в яку необхідно повернути системи координат. Якщо знайдена вершина знаходиться лівіше від центра осі, то робимо поворот за годинниковою стрілкою, інакше – проти.

Отже, зробивши поворот, ми отримали нову систему координат, в якій і будемо шукати наші відхилення.

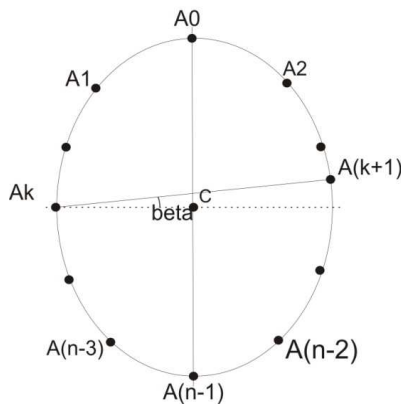


Рис. 2. Визначення кута відхилення

3.3. Алгоритм визначення відхилень

Крок 1. Для визначення відхилень спочатку ми будемо проходити по всіх парах точок, починаючи з A_1 .

Крок 2.1. Для кожної пари A_k, A_{k+1} шукаємо відповідний кут відхилення β . Якщо $\beta > 3^\circ$, знайдено відхилення. Отже, одна із точок A_k або A_{k+1} знаходиться у невірному положенні (рис. 2). Тепер необхідно знайти, яка саме із них: A_k чи A_{k+1} .

Крок 2.2. За вірність розміщення точки візьмемо її рівновіддаленість відносно вершини A_0 та наступної під нею точки. Тобто знайдемо різниці відстаней між вершиною A_0 та наступною точкою.

$$d_1 = \left| \sqrt{(A_0.x - A_k.x)^2 + (A_0.y - A_k.y)^2} - \sqrt{(A_k.x - A_{k+2}.x)^2 + (A_k.y - A_{k+2}.y)^2} \right|,$$

$$d_2 = \left| \sqrt{(A_0.x - A_{k+1}.x)^2 + (A_0.y - A_{k+1}.y)^2} - \sqrt{(A_{k+1}.x - A_{k+3}.x)^2 + (A_{k+1}.y - A_{k+3}.y)^2} \right|.$$

Якщо $d_1 > d_2$, то точка A_k знаходиться у вірному положенні. Отже, ми розраховуємо координати віртуальної точки A_{k+1} , тобто там, де вона мала б знаходитись:

$$A_v.y = A_k.y,$$

$$A_v.x = 2 * A_1.x - A_k.x.$$

Тепер ми можемо розрахувати довжину нашого відхилення.

$$\lambda = \sqrt{(A_{k+1}.x - A_v.x)^2 + (A_{k+1}.y - A_v.y)^2}.$$

Інакше, якщо $d_1 < d_2$, то точка A_{k+1} знаходиться у вірному положенні. Точка A_k повинна мати координати:

$$A_v \cdot y = A_{k+1} \cdot y,$$

$$A_v \cdot x = 2 * A_1 \cdot x - A_{k+1} \cdot x.$$

Остаточно маємо довжину відхилення:

$$\lambda = \sqrt{(A_k \cdot x - A_v \cdot x)^2 + (A_k \cdot y - A_v \cdot y)^2}.$$

Отже, ми отримали довжину та кут відхилення для нашої пари точок A_k, A_{k+1} . Для наочності будемо позначати на зображенні координати шуканої точки.

Крок 2.3. Повторимо кроки 2.1 та 2.2 для пар A_{k+2}, A_{k+3} доти, доки $k < n - 1$.

У результаті виконання кроку 2 ми отримали всі можливі відхилення, окрім крайньої верхньої та нижньої точок.

Крок 2.* Окремо проводимо крок для пари A_o, A_{n-1} . *Крок 2** майже повністю аналогічний крокові 2, окрім кількох зауважень. По-перше, потрібно поміняти x та y місцями (у кроці 2 ми перевіряли точки зліва направо, щоб у даному випадку перевірити зверху вниз, потрібно просто поміняти осі). По-друге, при обчисленні вірності тієї чи іншої точки використаємо такі формули:

$$d_1 = \left| \sqrt{(A_0 \cdot x - A_1 \cdot x)^2 + (A_0 \cdot y - A_1 \cdot y)^2} - \sqrt{(A_0 \cdot x - A_2 \cdot x)^2 + (A_0 \cdot y - A_2 \cdot y)^2} \right|,$$

$$d_2 = \left| \sqrt{(A_{n-1} \cdot x - A_{n-2} \cdot x)^2 + (A_{n-1} \cdot y - A_{n-2} \cdot y)^2} - \sqrt{(A_{n-2} \cdot x - A_{n-3} \cdot x)^2 + (A_{n-2} \cdot y - A_{n-3} \cdot y)^2} \right|.$$

Там, де різниця менша, там і буде вірною точка. Далі все аналогічно крокові 2.

Крок 3. Ми отримали наш результат, тепер повертаємось назад до старої системи координат. При подальшому використанні віртуальних точок їх також необхідно привести до старої системи координат (повернути систему на кут α у протилежний бік).

На цьому етапі ми отримали всі основні дані про відхилення, що цікавлять користувача.

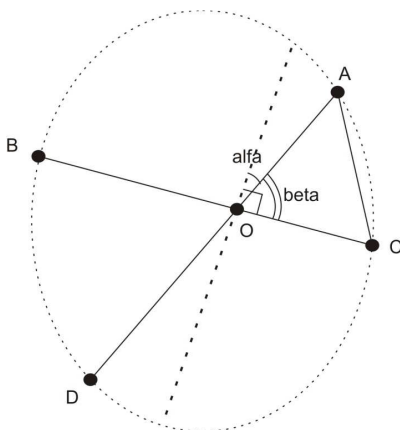


Рис. 3. Визначення відхилення для заданих контурних точок

тий.

3.4. Алгоритм визначення відхилення для заданих контурних точок

Як бачимо, в попередньому алгоритмі ми брали за аксіому, що точка має вірне положення, коли вона розташована на тому ж рівні, що й протилежна контурна точка, а вісь координат стандартна і визначається за положенням центральної точки C . Тепер припустимо, що користувач хоче сам задавати ті точки, між якими він прагне знайти відхилення. Отже, маємо таку задачу.

Користувач задає 4 точки (2 осі), між якими необхідно знайти відхилення (рис. 3). За вірність розташування осей візьмемо їх взаємну ортогональність. Щоб уникнути двозначності розв'язку, нехай користувач буде вказувати, яка з осей знаходиться у вірному положенні. Алгоритм вирішення такої задачі також достатньо простий.

Алгоритм

1. Спершу потрібно знайти точку перетину цих осей, щоб визначити новий центр координат. Позначимо її O . Задані точки позначимо відповідно A, B, C, D (рис. 3). Нехай вісь BC задана користувачем як вірна. Тоді задача зводиться до пошуку кута відхилення осі AD відносно BC . Позначимо шуканий кут α .

2. Оскільки ми маємо координати усіх заданих точок, а також координати точки O (як точки перетину прямих), нам необхідно лише знайти кут β .

3. А потім, беручи до уваги перпендикулярність розташування прямих у вірному випадку, кут $\alpha = 90 - \beta$. Кут β можна знайти за теоремою косинусів (оскільки нам відомі вершини A, O, C і сторони цього трикутника).

Таким чином, ми можемо знайти шукане нами відхилення.

4. Обґрунтування складності

Провівши аналіз даного алгоритму, очевидним є те, що його складність лінійна. Нехай N – кількість вхідних даних, тобто кількість точок, з якими ми працюємо. Ми переходимо до нової системи координат і повертаємось назад, тобто виконуємо $2N$ операцій. Окрім того, щоб знайти кут, на який потрібно повернути вісі, ми перебором будемо шукати найближчу з осей серед наших точок. Оскільки точки розташовані послідовно, ми знайдемо крайню точку осі максимум за $N/4$. На даний момент складність складає $2N + N/4$. Далі робимо перевірку всіх пар точок, тоді складність = $2N + 3N/4$. В найгіршому випадку відхилення є між кожною з пар, тому знаходження повної інформації по відхиленнях додає ще $N/2$ операцій. Таким чином, складність алгоритму при найгірших обставинах становитиме $O(N) = 3N + N/4$, тобто складність лінійна.

5. Практична частина

Для роботи з відеозображеннями ми використовуємо бібліотеку `OpenCv` [4]. Тут знаходяться всі потрібні нам функції для реалізації нашого алгоритму. Програма написана на мові `C++` з використанням програмного середовища `Visual Studio`.

Основними вхідними даними програми є відеопоток, з якого ми вилучаємо кадри для подальшої обробки. Окрім того, користувач може вручну задати 4 точки, для яких будуть шукатися відхилення. В залежності від потреб користувача вихідними даними є загальні відхилення, спеціальні відхилення (на 4 заданих точках), довжини всіх можливих контурів та осей. Окрім того, для наочності на вхідному відео виділяються знайдені точки та знайдені відхилення. На вихід також подається зображення з нумерацією точок, відображенням контуру та інших даних, які дозволять користувачу наочно слідкувати за роботою програми.

6. Висновки

У роботі представлено розробку ефективних інструментальних засобів (бібліотека алгоритмів) для обробки відеозображень із застосуванням в медицині (мануальна терапія та хірургія). Використовуючи звичайні технічні можливості сучасних комп'ютерів, можна вирішувати не лише задачі в медицині, але й великий клас задач у самих різноманітних областях. Беручи до уваги швидке зростання потужності сучасного обладнання, ми матимемо змогу використовувати їх для більш складних задач.

СПИСОК ЛІТЕРАТУРИ

1. Жариков О.Г. Экспертные системы в медицине / О.Г. Жариков, А.А. Литвин, В.А. Ковалёв // Медицинские новости. – 2008. – №10. – С. 15 – 18.
2. Близкая О. Разработка методов и алгоритмов обработки медицинских изображений с использованием методов искусственного интеллекта [Электронный ресурс] / О. Близкая. – Режим доступа: <http://masters.donntu.edu.ua/2005/kita/blizkaya/diss/index.htm>.
3. Кручинин А.Ю. Оптимизация систем распознавания образов реального времени / А.Ю. Кручинин // Автоматизация в промышленности. – 2010. – № 10. – С. 6 – 9.
4. Bradski G. Learning OpenCV; Computer Vision with the OpenCV Library / G. Bradski, A. Kaehler. – Cambridge: O'Reilly Media Publisher, 2008. – 577 p.

Стаття надійшла до редакції 18.03.2013