УДК 519.6

В. А. Турчина*, О. Ю. Лебідь, Є. В. Козаченко[*]**
**Дніпропетровський національний університет імені Олеся Гончара*
***Академія митної служби України*

МЕТОД ОЦІНЮВАННЯ ЯКОСТІ ІНТЕРФЕЙСУ КОРИСТУВАЧА СИСТЕМ ДИСТАНЦІЙНОГО НАВЧАННЯ

Розроблено метод оцінки якості інтерфейсу користувача систем дистанційного навчання, що враховує фізіологічні та психологічні аспекти сприйняття візуальної інформації людиною. Розроблено програмний продукт, яким можна вільно користуватися для оцінки якості власних інтерфейсів користувача.

Разработан метод оценки качества пользовательского интерфейса систем дистанционного обучения, учитывающий физиологические и психологические аспекты восприятия визуальной информации человеком. Разработан программный продукт, которым можно свободно пользоваться для оценки качества собственных интерфейсов пользователя.

A method for assessing the quality of the user interface of distance learning systems, taking into account the physiological and psychological aspects of perception of visual information man is elaborated. The software product which can be freely used for assessing the quality of their user interfaces is developed.

Ключові слова: інтерфейс користувача, система дистанційного навчання, фіксації погляду.

Постановка проблеми. Системи дистанційного навчання стають все більш популярними в умовах невпинного росту кількості Інтернет користувачів, рівня урбанізації та необхідності отримання якісної освіти.

Сьогодні за допомогою систем дистанційного навчання проходять навчання мільйони людей по всьому світу. Для будь-якого вищого навчального закладу одним з показників його високого рівня стає наявність системи дистанційного навчання, яку можна оцінити за якістю знань, що вона надає та по якості інтерфейсу системи, що впливає на засвоєння інформації.

Отже, якість інтерфейсу користувача системи дистанційної освіти є важливим фактором, що впливає на якість системи дистанційної освіти в цілому.

Під час створення системи дистанційної освіти розробник приділяє багато уваги питанням технологій, даних, розробки моделі системи, що навчає тощо, проте дуже рідко звертає увагу на те, що інтерфейс системи сильно впливає на якість та швидкість засвоєння інформації. Тому розробка методу, що дозволяє оцінити якість інтерфейсу користувача є дуже актуальною задачею не тільки для розробників систем дистанційної освіти.

Окрім розробки методу необхідно розробити систему оцінки якості інтерфейсу користувача, яка могла б давати відповідь на питання щодо якості розробленого інтерфейсу. Зважаючи на ріст чисельності користувачів Інтернет та збільшення якості Інтернет послуг в Україні доцільно розробити Інтернет-сервіс, який був би доступним і міг швидко надавати розробникам послуги по оцінюванню якості інтерфейсу користувача.

Аналіз досліджень та публікацій. Поява eye-tracking пов'язана з дослідженнями руху погляду, починаючи з 1879 року, коли Луїс Еміль Жаваль зробив спостереження, що при читанні не відбувається гладкого ковзання очами по тексту, як передбачалося раніше, а відбувається серія коротких зупинок, названих фіксаціями, і швидкий стрибкоподібний рух очей [2]. Інформацію про навколишній світ ми візуально сприймаємо тільки через фіксації. Це відкриття порушило важливі питання про процес читання, які досліджувалися на початку 1900-х років.

З тих пір і дотепер вчені намагалися визначити як впливають рухи очей на зір у цілому та на сприйняття зображень [1].

У 1950 р. Альфред Ярбус написав про взаємозв'язок між фіксаціями погляду й зацікавленістю людини в певному об'єкті. Рухи погляду відображають людські процеси мислення, тому думка респондента випливає, певною мірою, із записів рухів погляду. Тобто можна легко визначити, які елементи привертають увагу спостерігача й, отже, його думки, в якому порядку і як часто.

До появи сучасної комп'ютерної техніки вченим доводилося досліджувати рухи очей на піддослідних (наприклад, дослідження Edmund Huey). З появою сучасних комп'ютерних технологій ай-трекінг вийшов на новий рівень. Тепер поведінку піддослідних моделює комп'ютер. Проте, ці методи мають досить високу алгоритмічну складність, і, крім того, не досить високу швидкість роботи.

У 1980 р. М. Джаст і П. Карпентер сформулювали важливу гіпотезу сили взаємодії погляду й думки, відповідно до якої немає істотних відмінностей між тим, на чому зафіксований погляд і тим, що обмірковує людина. Дана гіпотеза сьогодні дуже часто розглядається як основа для початку досліджень у сфері eye-tracking.

Сьогодні існують алгоритми комп'ютерного моделювання ай-трекінгу, тобто комп'ютерні програми, які можуть без використання спеціальної апаратури та без допомоги піддослідних змоделювати рухи очей, проте вони не можуть гарантувати точність. По-перше тому, що деякі з саккад є свідомими, по друге – тому, що не усі маркери уваги (точки фіксації) можна знайти за допомогою обчислювальної техніки та програм.

© *В.А. Турчина, О.Ю. Лебідь, Є.В. Козаченко, 2010*

Іншим підходом до розв'язання задачі оцінювання інтерфейсу чи зображень у цілому є метод експертних оцінок. Очевидно, що такий підхід є занадто дорогим для такої задачі, бо для нього необхідна експертна група та приміщення і занадто довгим (при великих експертних групах процес оцінювання може займати до місяця).

Постановка задачі. Кожна людина сприймає візуальну інформацію, а значить і інтерфейс системи дистанційної навчання суб'єктивно, тому необхідно оволодіти знаннями щодо факторів, що впливають на сприйняття інформації людиною і розробити метод оцінювання інтерфейсу користувача з метою покращення сприйняття інформації учнями систем дистанційного навчання.

Eye-tracking дослідження, проведені Якобом Нільсеном [5], показують, що користувачі читають вміст, який нагадує форму F, а це означає, що читання починається з лівої верхньої частини веб-сторінки, і далі вона рухається трохи вниз, починаючи знову зліва (рис. 1).

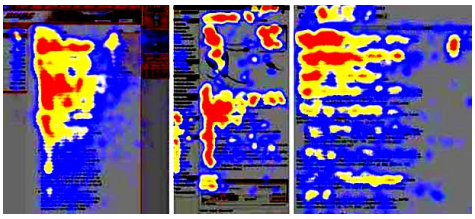


Рис. 1. Рух ока при читанні вмісту сайту [5]

Нільсен також заявляє про наслідки читання за схемою: користувачі не читатимуть вміст веб-сторінки по словах, вони витягнуть важливі параграфи, жирний текст, тощо; перші два параграфи повинні містити найбільш важливу інформацію; підрубрики і списки виділяються від регулярних параграфів.

Було досліджено, що інформація розміщена у пропорціях чисел Фібоначчі, «золотого перетину» чи на уявній синусоїді сприймається краще, ніж просто блок інформації через те, що за таких умов інформація набуває більш природного вигляду (рис. 2).

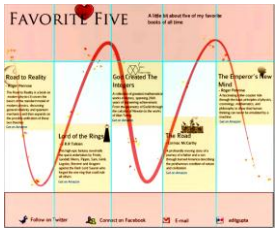


Рис. 2. Приклад розташування інформації на уявній синусоїді [3]

Розглянемо психологічні аспекти сприйняття. Уся інформація, що сприймається має або знаковий або аналоговий вираз. Сенсорні системи сприйняття людини утворюють внутрішню модель світу в 3 формах: звуках, образах і відчуттях.

У сенсорному сприйнятті знакової інформації існують дві системи – зорова і слухова. Механізми «дешифрування» знакової інформації можуть задіяти будь-яку систему сприйняття або всі три. Іншими словами, слово «яблуко» укладає до собе зміст форми, кольору, ваги, структури, смаку, запаху, звуку і таке ін.

Увага людини дивна та складна система, тому досі не існує обґрунтованої і точної математичної моделі уваги, а всі спроби хоч якось наблизитись до мети зводяться до моделювання фізіологічних процесів сприйняття, але людина – це не просто організм, це, у першу чергу, психологічний індивід, якому для сприйняття інформації необхідно звернути на неї увагу.

Шляхом лабораторних досліджень вченими було встановлено, що людина звертає увагу на деякі сектори зображення першочергово, а деякі не помічає зовсім. Те, що людина побачить першим може привернути її увагу і змінити хід її роботи, тобто важливо виявити ті місця, які людина побачить у першу чергу для того, щоб контролювати чи спробувати спрогнозувати дії, які зробить людина.

Сприйняття сайту має свої межі і знаходиться в залежності від особливостей механізму нервової системи людини.

Читання сайту відноситься до візуального сприйняття інформації. При цьому на те як людина помітить, прочитає, запам'ятає інформацію на сайті впливають відразу декілька факторів:

- кольорова гама сайту;
- завантаженість сторінки інформацією і графікою;
- кегль шрифту;
- контрастність тексту;
- геометрична «спрямованість» дизайну;
- наявність маркерів уваги.

Таким чином, на сприйняття інформації людиною впливає багато факторів, не залежно від того яке джерело. При цьому сприйняття – це суб'єктивний процес, тому оцінювати якість сприйняття зручніше засобами теорії нечітких множин та нечіткої логіки.

Розглянемо фізіологічні аспекти сприйняття. Оскільки людина сприймає сайт як візуальний об’єкт, то розглянемо лише ті інструменти, які відповідають за візуальне сприйняття, тобто зорову систему [4].

Зорова система починається з ока людини. Сітківка є аналогом фотосенсора фотоапарату, перероблюючи світло в набори електричних імпульсів та виконуючи первісну обробку зорової інформації, ще до «відправлення» її в зоровий відділ мозку.

На сітківці під двома шарами нервових клітин знаходяться зорові фоторецептори, які діляться на рецептори, що слабко залежать від ступеня освітлення (палички) – вони реагують на світло що надходить до ока, та колбочки, які реагують на більш сильне освітлення, їх більше, проте вони довше активізуються.

Розглянемо тільки колбочки. Колбочки відповідають за денний зір. Вони розділяються на 3 типи за кольорами, які сприймають (червоний, зелений і синій). Жовтий сприймається відразу двома видами колб, від того він значно більш інтенсивний, ніж синій. Через це генерується більш потужний електричний імпульс і жовтий нам здається більш яскравим і насиченим, ніж синій, на тому самому зображенні (цей факт перетинається з психологічним аспектом сприйняття описаним вище).

Усі фоторецептори об’єднуються у сектори, їх можна уявити як пікселі матриці фотоапарата. Ці сектори відповідають не лише за сприйняття інформації про колір, вони також виконують операції детекції руху та локалізують лінії на зображені.

У процесі сприйняття інформації сітківка, у першу чергу, оцінює яскравість сприйнятого кольору. Чим вище інтенсивність, тим більш яскравим здається колір. Однак при одній і тій же інтенсивності деякі кольори здаються більш яскравими.

Колірне коло Ньютона (рис. 3) відповідає максимальній насиченості кольору. Кольори розташовані так, щоб підкреслити деякі закономірності в їх сприйнятті. А саме, кожен колір має свій компліментарний колір, який займає в колірному колі діаметрально протилежну позицію. Суміш компліментарних кольорів, узятих за певним співвідношенням, утворює білий або сірий колір. Пари компліментарних кольорів є кольорами-антагоністами, оскільки вони анулюють вплив один одного на зорову систему. Приклади компліментарних кольорів: синій і жовтий, червоний і синьо-зелений, зелений і пурпурний, причому пурпурний лежить поза колірного кола, оскільки не має своєї колірної хвилі, а є відчуттям, що виникає при змішуванні спектрально-чистих кольорів.

Лабораторно доведено, що перепади яскравості та кольору сітківка сприймає виключно на границях перепаду. Цей факт добре демонструє ілюзія Маха (рис. 4).

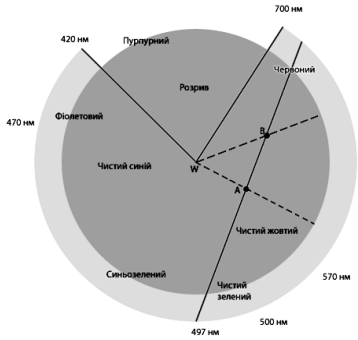


Рис. 3. Колірне коло Ньютона [4]

Границі між сусідніми по колірному колу кольори мають схильність сприйматися свідомістю як менш важливі, менш різкі.

Таким чином, ще до початку обробки зорової інформації мозок отримує від сітківки інформацію про лінії та контури на зображенні.

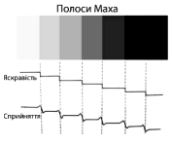


Рис. 4. Ілюзія Маха [4]

З властивостей роботи сітківки вченим вдалося вивести деякі закони сприйняття кольорів та яскравості: Вебера, Фехнера та Стівена.

Відносно чітке зображення око може сприйняти за 2.5 секунди (0,5 секунди на локалізацію і 2 секунди на точне налаштування) але при куті зору лише 10°. Можливість чітко бачити об’єкти, що знаходяться у більшому полі зору забезпечується завдяки зоровим саккадам, тобто рефлекторним переривчастим рухам очей, які переводять погляд з одного об’єкта на інший.

Сітківка працює виключно завдяки саккадам, бо спочатку була спроектована на розпізнавання рухів. Саккади бувають свідомі (коли можна свідомо перевести погляд на об’єкт) та несвідомі (відбуваються рефлекторно): прослідкувати за об’єктом, що перемістився; периферійним зором помітити натяк на

психологічний об’єкт (перетин з психологічним аспектом сприйняття); периферійним зором помітити візуально активний об’єкт (контрастні об’єкти, великі об’єкти).

Несвідомі саккади можна відловлювати, тобто змоделювати чи спрогнозувати. Очевидно, якщо на зображенні є людське обличчя, яке знаходиться у полі периферійного зору, то наступна саккада буде націлена на область з обличчям. Так само можна оцінити кольори: якщо на зображенні у полі периферійного зору є яскравий об’єкт на фоні компліментарного кольору, то саккада буде направлена на вивчення цього об’єкта.

Такий самий ефект відбувається і з контурами зображення. Так зображення з багатьма вигинами контуру притягує саккади, а значить і увагу, більше ніж зображення з точними геометричними контурами та прямими лініями. Крім того витіюватий контур має кращий психологічний ефект для запам’ятовування.

Очевидно, що психологічні та фізіологічні аспекти сприйняття тісно перетинаються між собою, тому для досконалої оцінки якості інтерфейсу ефективно було б оцінювати одразу два аспекти.

Зрозуміло, що на рухомі об’єкти (наприклад, анімовані картинки) користувач зверне увагу в першу чергу, тому виключимо їх із розгляду і обмежимося статичним зображенням.

Опишемо більш детально основні кроки реалізації методу.

Крок 1. Першим кроком методу є детекція переважаючого кольору сайту, оскільки, коли людина бачить сайт вперше, декілька секунд очі розфокусовані, вона не бачить деталей і контурів, вона бачить виключно кольори.

Для оцінки переважаючого кольору можна використовувати два підходи:

- «розмиття» зображення до отримання ефекту розфокусованих очей (наприклад, за допомогою фільтра Гауса);
- сильна пікселізація зображення з подальшим аналізом отриманих пікселів.

Слід зазначити, що для отримання придатного для оцінювання кольору розмиття необхідно використовувати фільтр Гауса з дуже великою дисперсією, що дає велике вікно фільтра та подовжує час виконання операції. Тобто час роботи алгоритму великий, і тому він не придатний для використання.

Алгоритм пікселізації працює на декілька порядків швидше ніж алгоритм фільтра Гауса, що дозволяє застосувати його для первісної обробки зображення.

Як і в більшості алгоритмів обробки зображення, в алгоритмі пікселізації зображення рухаємо вікно обробки по зображенню. На відміну від методу Гауса, де рух вікна відбувається з кроком в один піксель, у даному методі щоразу зміщуємо вікно на відстань, яка дорівнює довжині вікна, що значно економить час. На кожному кроці необхідно обчислити середній колір вмісту вікна обробки. Для цього:

- в кожному пікселі вікна обробки отримаємо колір, тобто обчислюємо колір зображення в конкретній точці;
- розкладаємо отриманий колір на складові за гамою RGB;
- додаємо отримані компоненти кольору до суми усіх кольорів вікна обробки (також покомпонентно);
- після того, як пройдемо крізь усі пікселі вікна обробки обчислимо середні значення компонент кольору:

$$\rho = \frac{\sum_{i=0}^{n-1} R_i}{T},\ \gamma = \frac{\sum_{i=0}^{n-1} G_i}{T},\ \beta = \frac{\sum_{i=0}^{n-1} B_i}{T}, \quad (1)$$

де ρ, γ, β – це середні значення кольору у вікні обробки покомпонентно відповідно червоного, зеленого та синього компонентів; T – загальна кількість оброблених пікселів у вікні; R_i, G_i, B_i – відповідні компоненти кольору кожного з пікселів вікна, n – кількість пікселів.

У результаті роботи алгоритму отримаємо середнє значення кольору зображення у вікні обробки. Тепер можна зафарбувати вікно обробки в отриманий колір, змістити його і перейти до обробки наступного блоку пікселів. У результаті отримаємо набір прямокутників пофарбованих у різні кольори (рис. 5).



Рис. 5. Приклад роботи алгоритму пікселізації

Для реалізації було обрано розмір вікна обробки 100x100 пікселів. Отримаємо n*10 квадратів, де n – результат ділення висоти зображення на 100. Обчислюємо середній колір усіх квадратів і визначаємо переважаючий колір на зображенні. Цей колір, у подальшому, можна проаналізувати і встановити чи сприятливий він для користувача.

Крок 2. Людські обличчя відносяться до психологічних атракторів уваги, тому в методі оцінки інтерфейсу користувача доцільно знайти та локалізувати обличчя для подальшого аналізу їх розташування на зображенні і оцінки впливу на розподіл уваги користувача.

Локалізація облич – це комп’ютерна технологія, що виявляє людські обличчя будь-якого розміру на зображенні, відокремлюючи риси обличчя від непотрібного шуму (дерев, будівель тощо) [6].

До методів локалізації облич відносяться: локалізація облич як задача класифікації шаблону, локалізація облич на контрольованому фоні, локалізація облич за кольором.

У даній роботі відбулась свідома відмова від використання методів локалізації облич, що базуються на нейронних мережах, оскільки такі мережі потребують великої навчальної вибірки і, крім того, мають досить складну структуру, через що дуже довго навчаються.

Алгоритм для локалізації облич базується на праці [8]:

- відшукати можливі області, де може бути обличчя, керуючись міркуваннями про колір шкіри;
- бінаризувати зображення в тих областях, де може знаходитись обличчя;
- зіставити отримані області із бінарним шаблоном облич, якщо шаблони співпадають, то маємо обличчя на зображенні.

Для реалізації алгоритму необхідно використати спосіб кодування RGB інформації – YC_BC_R. Фактичні кольори відображаються залежно від фактичного RGB і використовуються для відображення сигналу [10].

Для переведення зображення з кольорового простору RGB до YC_BC_R, необхідно обчислити наступні значення для кожного пікселя:

$$Y = K_R R + (1 - K_R - K_B)G + K_B B, \tag{2}$$

$$C_B = \frac{1}{2} \cdot \frac{B - Y}{1 - K_B}, \tag{3}$$

$$C_R = \frac{1}{2} \cdot \frac{R - Y}{1 - K_R}, \tag{4}$$

де $K_R = 0{,}114,\ K_B = 0{,}299$ – визначені константи.

Після конвертування зображення у YC_BC_R проходимо зображення, порівнюючи колір кожного пікселя з множиною порогових значень

$$Y = 0{,}5,\ C_B \in [-0{,}15; 0{,}05],\ C_R \in [0{,}05; 0{,}20]. \tag{5}$$

Якщо колір пікселю потрапляє в описані межі, то відмічаємо його, як «підозрілий» і ставимо на зображенні білу точку, інакше ставимо чорну точку та отримуємо карту (рис. 6).



Рис. 6. Карта кольорів, що можуть бути обличчям [9]

Тепер необхідно прибрати зайві білі точки, шум, тобто точки, які не мають багатьох сусідніх. Для цього використовуємо алгоритм ерозії [11], у результаті роботи якого прибираємо з карти кольорів зайвий шум, але із зменшенням розмірів «підозрюваних» на обличчя областей, що може ускладнити роботи на наступному кроці. Для того, щоб повернути початкові розміри областей застосуємо метод математичної морфології метод розширення [12]. Основний ефект дії методу полягає в тому, що межі об'єктів розширюються, а незаповнені проміжки у них зменшуються.

Отримали зображення, відфільтроване за кольором шкіри. Необхідно сегментувати зображення для локалізації обличчя. Для цього потрібно знайти області на зображенні з найбільшим вмістом білих пікселів. Обчислимо суми білих пікселів за кожним стовпцем і за кожним рядком. Місця з максимальною кількістю білих пікселів за стовпцями та рядками одночасно і будуть центром обличчя.

Для того, щоб визначити межі обличчя використовуємо обчислені суми білих пікселів. Межі обличчя починаються при відхиленні від суми білих пікселів, що відповідають центру обличчя, на деяке порогове значення. Відсіюючи всі значення, менші за порогове, визначаємо межі обличчя.

Крок 3. Локалізація тексту на довільному зображенні є нетривіальною задачею, оскільки на відміну від облич текст має складнішу структуру і майже не піддається шаблонізації [7].

Текст є об'єктом з високим вмістом прямих ліній та гострих кутів, тому для того, щоб знайти області, які гіпотетично можуть містити текст слід обробити зображення за допомогою алгоритмів пошуку границь. У результаті отримаємо набір областей, «підозрюваних» на вміст тексту. Далі кожну з цих областей зможемо обробити специфічним алгоритмом детекції тексту.

Після локалізації тексту необхідно визначити його читабельність. Для цього необхідно знайти колір самого тексту та колір фону, на якому він розташований.

Після локалізації тексту та з'ясування кольорів літер та фону, кольори треба оцінити, для чого рекомендовано використовувати прості методи оцінки кольорової різниці, різниці яскравості та контрастності. Усі ці методи оцінюють два кольори як розкладання у спектр RGB.

У роботі використано метод локалізації тексту, що базується на методі CED [13] (англ. Color edge detection – детекція границі кольору) та виділення рядків тексту. Причини такого вибору: складна і невідома структура фону зображення, що буде оброблятися, яка відкидає методи пошуку, котрі базуються на кольорах; можлива низька контрастність тексту, що шукається, відкидаюча методи, які базуються на виявлені контрастних

повторюваних структур; у більшості зображень, які оброблятимуться (сайтах) текст розташований консолідованими блоками по рядках, тому доцільно проводити класифікацію областей «підозрюваних» на текст за допомогою аналізу рядків.

CED-оператор є оператором виділення границь, який в обробці керується повним впливом колірного простору YIQ [14]. Сигнал I називається синфазним, Q – квадратурним. Конверсія в RGB і обернено здійснюється за наступними формулами:

$$R = Y + 0{,}956I + 0{,}623Q, \quad G = Y - 0{,}272I - 0{,}648Q,$$
$$B = Y - 1{,}105I + 0{,}705Q, \tag{6}$$

$$Y = 0{,}299R + 0{,}587G + 0{,}115B, \quad I = 0{,}596R - 0{,}274G - 0{,}322B,$$
$$Q = 0{,}211R - 0{,}522G + 0{,}311B, \tag{7}$$

де R, G, B – відповідають інтенсивності кольорів червоного, зеленого і синього; Y – складова якравості; I та Q – кольорорізницеві складові.

CED об'єднує вживані вище морфологічні оператори, але в комбінації із конвертацією зображення в YIQ, що дозволяє обробляти не тільки зображення у сірих відтінках, а й повно-кольорові зображення.

У подальших викладках використовується оператор Робертса [15].

Висока точність і здатність до видалення шумів – важливі вимоги для виявлення границь на кольорових зображеннях, так само, як і на сірих зображеннях. Тут традиційний оператор Робертса перетворюється в CED, який використовує YIQ систему кольору. Оператор CED описується наступним чином

$$CED = \sqrt{\delta_1^2 + \delta_2^2}, \tag{8}$$

де δ_1 та δ_2 визначаються

$$\delta_1 = Dis(i, j, i+1, j+1), \tag{9}$$

$$\delta_2 = Dis(i+1, j, i, j+1), \tag{10}$$

де $Dis(\cdot)$ – евклідова норма двох пікселів на зображенні в YIQ колірній системі.

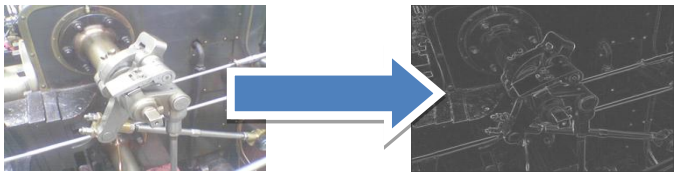


Рис. 7. Результат роботи оператора Робертса [15]

Постобробка важлива для сегментації тексту і фону в тих зображеннях, які обробили CED. Оскільки текстові лінії зазвичай горизонтальні, необхідно посилити горизонтальні краї зображення. Так, тут використовується оператор виділення границь, щоб виділити край зображення знову після того, як CED виділив вперше. Для цього використовується оператор Собеля [16].

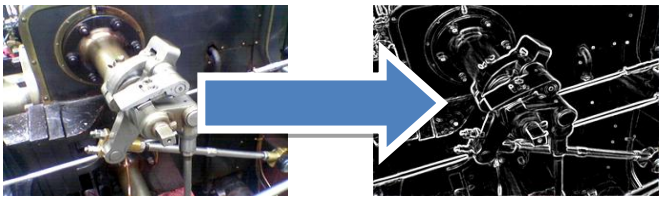


Рис. 8. Результат роботи оператора Собеля [16]

Таким чином, отримано бінарне зображення і можна локалізувати блоки, «підозрювані» на текст морфологічними методами.

Алгоритм для кроку локалізації тексту.

1. Вхідне зображення I_1 обробляється CED для отримання бінарного зображення I_2.
2. I_2 обробляється оператором Собеля і отримуємо бінарне зображення границь I_3.
3. I_3 обробляється морфологічними методами, і отримуємо бінарне зображення I_4 без шуму та з покращеними границями. Зважаючи на горизонтальну орієнтацію тексту рекомендовано використовувати відкриваючий, а потім закриваючий морфологічний оператор.
4. Після того, як обробка закінчена, слід застосувати важливі правила, щоб визначити деякі очевидні текстові блоки і видалити деякі очевидні нетекстові блоки. Для цього використовують як горизонтальні так і вертикальні проекції зображення I_4 та його щільність.

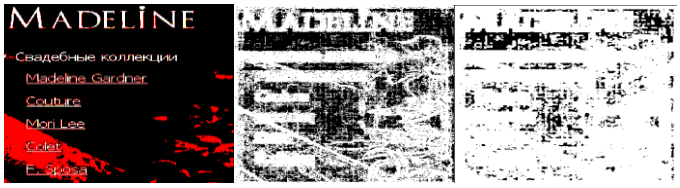
Позначимо через P_h – горизонтальну проекцію блоку зображення $m \times n$, а через P_v – вертикальну проекцію. Якщо не виконується нерівність

$$P_h > \mu_1 \text{ та } P_v > \mu_2, \tag{11}$$

де μ_1, μ_2 – порогові константи, то блок класифікується нетекстовим.

Якщо щільність блоку $m \times n$ менша за іншу порогову константу μ_3, то блок класифікується як нетекстовий. Якщо щільність блоку $m \times n$ більша за μ_3 та виконується нерівність (11), то блок класифікується як текстовий.

Таким чином процес локалізації тексту завершено (рис. 9).



а) б) в)

Рис. 9. Етапи обробки зображення за CED: а) вхідне зображення; б) після обробки CED; в) після застосування морфологічних операторів

Маємо набір блоків-кандидатів на присутність тексту. Необхідно визначити чи є в цих блоках текст.

Щоб прибрати зайвий шум, отриманий у ході обробки зображення оператором Собеля (шум з'являється через неоднорідність фону) застосуємо до сегментів, оброблених оператором Собеля методи математичної морфології, такі як ерозія та розширення (рис. 10).



Рис. 10. Результат обробки сегментів оператором Собеля та математичною морфологією

Тепер можна виділити рядки на зображенні. Задача виділення рядків зводиться до знаходження верхніх і нижніх граней рядків тексту, зображеного на початковому зображенні [17].

Алгоритм сегментації рядків ґрунтується на тому, що середня яскравість у зображеннях міжрядкових проміжках істотно нижче за середню яскравість зображень текстових рядків.

Спочатку для усіх піксельних рядків початкового зображення знаходимо їх середні значення яскравості

$$s_j = s_j(B) = \frac{1}{n}\sum_{i=1}^{n} b_{ij}, \tag{12}$$

де B – матриця яскравості точок вигляду

$$B = \begin{pmatrix} b_{11} & \dots & b_{1m} \\ \dots & \dots & \dots \\ b_{n1} & \dots & b_{nm} \end{pmatrix}, \tag{13}$$

а b_{ij} – значення яскравості конкретної точки $0 \le b_{ij} \le b^{\max}$, $b^{\max}$ – максимально можлива яскравість.

Обраховуємо середнє значення яскравості для всього зображення

$$s(B) = \frac{1}{m}\sum_{j=1}^{m} s_j(B). \tag{14}$$

Середня яскравість у міжрядкових проміжках тексту має бути невелика (в ідеальному випадку вона дорівнює нулю). Тому яскравість верхньої межі текстового рядка можна виразити через середню яскравість зображення

$$s^t = k^t s(B), \tag{15}$$

де k^t – коефіцієнт, причому $0 \le k^t \le 1$.

Аналогічно яскравість нижньої межі текстового рядка, також може бути виражена через середню яскравість усього зображення

$$s_b = k_b s(B), \tag{16}$$

де k_b – коефіцієнт, причому $0 \le k_b \le 1$.

Робота алгоритму сегментації рядків полягає у послідовному перегляді масиву середніх значень ($s_1,\dots,s_m$) і виявленні безлічі пар індексів (s_i^t, s_{b_j}) піксельних рядків, що відповідають верхній s_i^t і нижній s_{b_j} граням зображення i -го рядка, що задовольняють наступним умовам.

Умови верхньої межі текстового рядка. Початок текстового рядка або області стійкого підвищення яскравості фіксується, якщо

$$\begin{aligned}(s_{j-2} < s^t) \& (s_{j-1} < s^t) \& (s_j > s_b) \& (s_{j+1} > s_b) \& \\ \& (s_{j+2} > s_b) \& (s_{j+3} > s_b).\end{aligned} \tag{17}$$

Умови нижньої межі текстового рядка. Кінець області стійкого підвищення яскравості визначається, якщо

$$((s_j > s^t) \& (s_{j+1} < s_b)) OR ((s_{j+1} < s_b) \& (s_{j+2} > s_b) \& (s_{j+3} > s_b)). \tag{18}$$

У результаті формується безліч пар індексів верхніх і нижніх меж рядків. Різниця між цими індексами дає висоти текстових рядків. Проте такий алгоритм знаходить середню висоту кожного текстового рядка і «зрізає» символи, виступаючі по висоті за цю середню висоту.

Щоб уникнути цього, необхідно розширити знайдені межі: серед знайдених текстових рядків визначається рядок з мінімальною висотою $H_{\min}$ і, потім усі межі з кожного боку розширюються на величину $0{,}3H_{\min}$. Це не призводить до злиття рядків, оскільки міжрядкові інтервали тексту, як правило, більше ніж висота рядка.

Ті сегменти, які задовольняють наведеним вище умовам класифікуються як текст, інші видаляються з розгляду.

Отримали сегменти зображення, які містять текст і постала задача об'єднання їх за належністю до однакових блоків тексту [18].

Для цього будемо керуватися простим алгоритмом.

1. Оберемо перший необроблений сегмент, позначимо його як А та додамо його до списку блоків. Позначимо сегмент, як оброблений.

2. Знайдемо усі прилеглі до нього сегменти, позначимо їх, як оброблені та розширимо блок А значеннями прилеглих сегментів.

3. Рекурсивно повторимо крок 2 для сегментів, прилеглих до блоку А.

4. Якщо прилеглих сегментів не має, то позначимо блок А закритим, та перейдемо до кроку 1.

У результаті роботи цього алгоритму отримаємо блоки тексту.

Окрім локалізації тексту нам необхідно оцінити його читабельність. Для цього нам потрібно визначити колір фону, на якому розміщено текст, та значення кольору самого тексту.

Перш за все для обчислення значення кольору нам необхідно визначити де на виділеному текстовому блоці знаходиться текст, а де фон. І тут знову нам знадобиться зображення отримане в результаті обробки сегментів-кандидатів оператором Собеля та математичною морфологією (ерозією та розширенням). До цього зображення застосуємо пошук за шаблоном бччбб, де через б позначено білий піксель, а через ч – чорний. Зрозуміло, що в результаті обробки текстового блока оператором Собеля, ерозіїю та розширенням отримаємо точні контури тексту чорного кольору, фон, на якому знаходиться текст – білого кольору, сама літера знаходиться між контурами, що були отримані в результаті обробки операторами виділення границь. Шаблон пошуку: один піксель фону, два пікселі контуру та два пікселі літери. Координати, що відповідають правій границі шаблону – внутрішня частина літери, а лівій – фону. На зображенні визначаємо значення кольорів фону та переднього плану за отриманими координатами.

Для оцінки читабельності тексту використовують 3 фактори [19]:

– різницю кольорів:

$$dC = \max(R_1, R_2) - \min(R_1, R_2) + \max(B_1, B_2) - \min(B_1, B_2) + \max(G_1, G_2) - \min(G_1, G_2) \quad (19)$$

де R_i, G_i, B_i – відповідні складові 1-го та 2-го кольорів у гамі RGB;

– яскравісну різність за кольоровим простором YC_BC_R:

$$dB_{r_1} = \left| B_{r_1} - B_{r_2} \right|, \quad (20)$$

$$B_{r_1} = 0{,}299R_1 + 0{,}587G_1 + 0{,}114B_1, \quad (21)$$

$$B_{r_2} = 0{,}299R_2 + 0{,}587G_2 + 0{,}114B_2, \quad (22)$$

де R_i, G_i, B_i – відповідні складові 1-го та 2-го кольорів у гамі RGB;

– освітленість:

$$L_1 = 0{,}2126\left(\frac{R_1}{255}\right)^{2{,}2} + 0{,}7152\left(\frac{G_1}{255}\right)^{2{,}2} + 0{,}0722\left(\frac{B_1}{255}\right)^{2{,}2}, \quad (23)$$

$$L_2 = 0{,}2126\left(\frac{R_2}{255}\right)^{2{,}2} + 0{,}7152\left(\frac{G_2}{255}\right)^{2{,}2} + 0{,}0722\left(\frac{B_2}{255}\right)^{2{,}2}. \quad (24)$$

Якщо $L_1 > L_2$, то $dL = \frac{L_1 + 0{,}05}{L_2 + 0{,}05}$. Якщо $L_2 > L_1$, то $dL = \frac{L_2 + 0{,}05}{L_1 + 0{,}05}$.

Після того, як отримано оцінки читабельності тексту їх слід порівняти з константами.

Якщо $dC > 500$ & $dBr > 125$ & $dL > 5$, то текст вважають добре читабельним. Ці константи є емпіричними, і невеликим відхиленням значень оцінок від них у менший бік можна знехтувати.

Крок 4. Локалізація об'єктів з великим числом вигинів контуру. Окрім психологічно активних об'єктів «цікавими» для людського ока є і незвичайні об'єкти, які мають велику кількість вигинів контурів, тож у методі оцінки інтерфейсу користувача необхідно врахувати цей фактор.

Локалізація таких об'єктів є задачею досить схожою на задачу локалізації тексту, оскільки розв'язок тієї задачі також базується на виділенні контурів.

Крок 5. Геометрична направленість зображення має безпосередній вплив на психологічний стан користувача. Так, наприклад, зображення, яке більшістю об'єктів вказує вгору та вправо викликає у людини враження перспективності та прогресу, руху вперед, і, навпаки, зображення направлене вниз і вліво – з регресом та занепадом.

Крок 6. Отримані на попередніх кроках дані необхідно проаналізувати для того, щоб видати відповідь користувачу.

Кожен із блоків даних, отриманих на попередніх кроках, представляє собою набір із різнотипних не нормалізованих показників, таких, як, наприклад, позиції і розміри облич чи позиції, розміри та читабельність блоків тексту.

Оскільки інформації буде досить багато, доцільно аналізувати кожен із блоків даних, отриманих на різних кроках методу окремо, отримуючи окрему відповідь по кожному із блоків, а потім оцінювати відповіді по блокам.

Кожен із блоків даних, крім даних отриманих на першому кроці методу, має бути проаналізований наступним чином:

- перш за все слід виділити великі за розміром об’єкти (великі обличчя, великі блоки тексту), оскільки вони є впливовішими на сприйняття користувача;
- кожен із виділених об’єктів має бути проаналізованим щодо його розташування на зображенні (відомо, що найбільш впливовими на користувача є об’єкти розташовані у верхній та правій частинах зображення), тож виділимо тільки ті великі об’єкти, які «впливово» розташовані;
- тепер кожен із об’єктів має пройти відповідний аналіз (наприклад, для блоків із текстом слід оцінити їх читабельність).

У результаті отримаємо дані про впливовість кожного із блоків даних, отриманих у ході обробки зображення.

Представимо їх у вигляді лінгвістичних змінних, величин, множин та функцій належності [8]:

$$\varphi = \begin{cases} \varphi_1, \\ \dots, \\ \varphi_n, \end{cases} \qquad T = \begin{cases} T_1, \\ \dots, \\ T_n, \end{cases} \qquad \mu = \begin{cases} \mu_{\varphi_1}, \\ \dots, \\ \mu_{\varphi_n}, \end{cases} \tag{25}$$

де φ – сім’я лінгвістичних величин, T – набір нечітких величин, а μ – сім’я функцій належності.

Описавши ці дані і сформувавши набір правил логічного висновку можна видати відповідь користувачу щодо якості досліджуваного інтерфейсу.

Розглянемо приклад зображення з текстом та обличчям та локалізуємо обличчя на цьому зображенні (рис. 11).



Рис. 11. Результат локалізації обличчя

Проаналізуємо знайдені обличчя на зображенні. У результаті процедури локалізації обличчя знайдено одне обличчя. Обличчя знаходиться у центрі зображення, отже притягне увагу людини першочергово, і оскільки обличчя знаходиться безпосередньо в тексті, то це розташування притягне увагу до тексту. Крім того, обличчя є досить великим (14 % у ширину і 13 % у висоту відносно усього зображення).

Для подальшого аналізу зображення, та оцінки читабельності тексту проведемо процедуру локалізації тексту на зображенні та оцінку його кольору і кольору фону (рис. 12).



Рис. 12. Результат локалізації тексту

Проаналізуємо читабельність тексту: $dC = 765 > 500,\ dB = 255 > 125,\ dL = 21 > 5.$ Отже, маємо гарну читабельність тексту.

Висновки. Було розроблено метод оцінювання якості інтерфейсу користувача, який базується на оцінці психологічних та фізіологічних факторів, що впливають на сприйняття візуальної інформації людиною.

На основі розробленого методу створено програмний продукт, за допомогою якого можна проводити частину досліджень для отримання оцінки інтерфейсу користувача.

Програмний продукт має наступні функціональні можливості: проводити аналіз переважаючого кольору дистанційної системи, що навчає; локалізувати обличчя в інтерфейсі досліджуваної системи та оцінювати їх розташування і розміри відносно всього інтерфейсу; локалізувати текст в інтерфейсі досліджуваної системи і аналізувати його читабельність.

Надалі авторами планується повна реалізація запропонованого методу у вигляді програмного засобу для вільного користування.

Бібліографічні посилання

1. «Eye tracking» http://en.wikipedia.org/wiki/Eye_tracking
2. http://eyetracking.com.ua
3. «Applying Mathematics to Web Design» http://www.smashingmagazine.com/2010/02/09/applying-mathematics-to-web-design/
4. «Моделирование зрения» http://ymik.habrahabr.ru/blog/57679/
5. http://www.webdesignerdepot.com/2008/12/10-usability-tips-for-web-designers/
6. «Face detection» http://en.wikipedia.org/wiki/Face_detection
7. Datong Chen, Jean-Marc Odobez, Hervé Bourlard. Text detection and recognition in images and video frames. – IDIAP.: 2003. – 14 p.
8. **Ободан Н. І.** Моделі та методи систем штучного інтелекту. / Н. І. Ободан, В. А. Громов – Д., 2008. – 119 с.
9. http://blogs.msdn.com/b/coding4fun/archive/2010/03/24/9984015.aspx
10. http://en.wikipedia.org/wiki/YCbCr
11. http://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm
12. http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm
13. Yan Hao, Zhang Yi, Hou Zeng-guang, Tan Min. Automatic Text Detection In Video Frames Based on Bootstrap Artificial Neural Network And CED. – Plzen, Czech Republic.: 2003. – 6 p.
14. http://en.wikipedia.org/wiki/YIQ
15. Davis LS. A survey of edge detection techniques. Computer Graphics and Image Processing, vol 4, no. 3, 1975, pp 248-260
16. http://ru.wikipedia.org/wiki/Оператор_Собеля
17. http://mechanoid.narod.ru/misc/segmentator/
18. **Хорн Б.** Психология машинного зрения. / Б. Хорн, М. Минский, Й. Сараи и др. – М., 1978. – 337 с.
19. **Марр Д.** Зрение. Информационный подход к изучению представления и обработки зрительных образов. / Д. Марр – М., 1987. – 374 с.

Надійшла до редколегії 30.01.10