УДК 004.056.53; 004.051

Р. Кутельмах, Н. Павич, Х. Андрухів
Національний університет "Львівська політехніка",
кафедра програмного забезпечення

АЛГОРИТМІЧНІ ТА ПРОГРАМНІ ЗАСОБИ ОПРАЦЮВАННЯ ЗОБРАЖЕНЬ В ОПЕРАЦІЙНІЙ СИСТЕМІ MAC OS X

Запропоновано алгоритмічні та програмні засоби опрацювання зображень в операційній системі Mac OS X. Забезпечено можливість коригування колірних характеристик зображення, змінюючи контрастність, чіткість, насиченість тіней, інтенсивність теплих відтінків, а також усунення ефекту червоних очей. Показані результати тестування запропонованих засобів, які підтверджують їх ефективність.

Ключові слова: алгоритмічні засоби, програмні засоби, опрацювання зображень, операційна система Mac OS X.

The paper describes image processing software and algorithms approaches and for Mac OS X. The solution provides the opportunity to adjust image color characteristics by changing the contrast, sharpness, shadows, warmth and also "red eyes" effect removal. There are proposed techniques testing results, which proves these techniques approaches efficiency.

Key words: algorithmic approaches, software approaches, image processing, Mac OS X

Вступ

Сьогодні доволі часто у багатьох користувачів комп'ютерної техніки виникає задача опрацювання зображень, фотографій, передусім для покращення їх якості, для виправлення дефектів, для надання їм особливого стилю та вигляду. Тому актуальними є дослідження та конкретні розробки щодо створення алгоритмічних та програмних засобів опрацювання зображень.

Під час опрацювання зображень важливо, щоб програмний засіб ефективно розв'язував поставлену задачу та був простим у використанні. Зокрема такі задачі можна розв'язувати в операційній системі Mac OS X. Оскільки програмний інтерфейс самої операційної системи Mac OS X для більшості користувачів є зовнішньо привабливим, зручним, зрозумілим, легким в освоєнні та користуванні, то актуальність розробки програмного продукту для опрацювання зображень в цій операційній системі є цілком обґрунтованою.

Огляд літературних джерел та публікацій

Операційна система Mac OS X є доволі популярною у користувачів і, згідно із статистикою [1], продовжує набувати все більшої популярності. Компанія-виробник Apple надає можливість розробникам представляти свої програмні продукти мільйонам користувачів по всьому світу. Компанія Apple пропонує безкоштовне програмне середовище розробки програм Xcode, що об'єднує потужний набір інструментів для створення сучасних високопродуктивних програм, які є доступними в App Store [2]. Користувачам потрібно лише ввести ключові слова, обрати зі списку запропонованих продуктів бажаний і завантажити та автоматично встановити його на свій пристрій, використовуючи онлайн-магазин App Store.

Основними компонентами Mac OS X є [3]:

- підсистема з відкритим кодом – Darwin (ядро Mach, набір утиліт BSD);
- середовище програмування Core Foundation (Carbon API, Cocoa API і Java API);
- графічне середовище Aqua (QuickTime, Quartz Extreme і OpenGL);
- технології CoreImage, CoreAudio і CoreData.

Нині відомо низку програмних продуктів, що дають змогу обробляти зображення в Mac OS X. Найвідоміші програми : InPaint, uPhotos, iSplash Colors, Snapheal, ACDSee Photo Flash [2].

InPaint – програмний продукт, призначений для видалення з фотографії небажаних об'єктів. Програма достатньо швидко обробляє фотографію.

uPhotos – програмний продукт, з максимально спрощеним інтерфейсом, застосовує різні до завантаженого зображення. Стилі встановлюються за допомогою відповідних елементів управління на панелі інструментів. Засіб є достатньо простим, швидким у обробці фото, проте має дуже обмежений набір можливостей та недостатньо привабливий інтерфейс користувача.

iSplash Colors – програмний продукт, що дозволяє редагувати фотографії. Його основною особливістю є те, що він дає змогу застосовувати різні стилі до фотографії та певним способом зафарбовувати окремі ділянки фотографії.

ACDSee Photo Flash – програмний продукт, який призначений для зміни освітленості та контрастності зображення. Продукт є простим у використанні. Для зміни контрастності фотографії достатньо за допомогою повзунка встановити потрібне значення і одразу отримати результат.

Shapheal – один з найвідоміших програмних продуктів обробки зображень для Mac OS X, дозволяє застосовувати різні стилі, ефекти до зображення, наприклад, чорно-білий, відтінки сірого, яскравість, чіткість, а також видаляти небажані об'єкти з певної ділянки.

Кожен із розглянутих програмних продуктів здебільшого не забезпечує вимоги пересічного користувача, вимагає спеціальних навиків для роботи з зображеннями, деякі з них мають складний інтерфейс користувача.

Постановка задачі

Постановка задачі: розробити максимально просте у використанні програмне забезпечення для опрацювання зображень в Mac OS X з нестандартними компонентами інтерфейсу користувача. Програма повинна вміти застосовувати такі ефекти до зображення, як: зміна чіткості, яскравості, контрастності, інтенсивності тіней, інтенсивності світла, інтенсивності «теплих» відтінків, насиченості кольорів, зміна кута нахилу, інверсний поворот зображення, видалення об'єктів з певної області зображення.

Виклад розв'язання поставленої задачі

Компанія Apple пропонує безкоштовне програмне забезпечення з достатнім набором інструментів для створення якісних програмних додатків для Mac OS X. Водночас вона пропонує платформи, на базі яких можна створити програми різного характеру та призначення. Деякі платформи є вбудованими в саму операційну систему.

Carbon – це процедурний прикладний програмний інтерфейс (API), виданий компанією Apple Inc. для операційної системи Mac OS. Середовище Carbon засновано на галузевих стандартних мовах програмування C та C++. Ззовні майже неможливо знайти відмінності між застосунками Carbon та Cocoa.

Cocoa [4] – рідний об'єктно-орієнтований прикладний програмний інтерфейс (API) для операційної системи Mac OS X виробництва компанії Apple. Це один із п'яти основних API, доступних для Mac OS X (Cocoa, Carbon, Toolbox (для роботи старих додатків Mac OS 9), POSIX і Java.

Cocoa складається в основному з двох бібліотек об'єктів Objective-C, фреймворків :

- Foundation Kit, часто просто Foundation, уперше з'явився в OpenStep. В Mac OS X він заснований на основі Core Foundation. Foundation являє собою об'єктно-орієнтовану бібліотеку загального призначення, яка забезпечує роботу з рядками та значеннями, контейнери та ітерацію по них, розподілені обчислення, цикли опрацювання повідомлень та інші функції, не прив`язані безпосередньо до графічного інтерфейсу.
- Application Kit або AppKit походить по напряму від NeXTSTEP Application Kit. Він містить код, за допомогою якого програми можуть створювати графічний інтерфейс та взаємодіяти з ним. AppKit побудований на основі Foundation.

Ключовий елемент архітектури Cocoa – це модель переглядів (views). Зовнішньо вона організована як звичайний фреймворк, але реалізована з використанням PDF для усіх операцій малювання, схожого з PostScript. Крім того, це автоматично надає можливість виведення будь-якого перегляду на друк. Оскільки Cocoa обробляє обрізку, прокрутку, масштабування та інші типові задачі відображення графіки, програміст звільняється від необхідності реалізовувати базову інфраструктуру і може зосередитися на унікальних аспектах розроблюваної програми.

Програми, що використовують Cocoa, зазвичай розробляють за допомогою середовища розробки Apple Xcode (у минулому називався Project Builder) і Interface Builder з використанням мови Objective-C.

Середовище розробки Cocoa також доступне і під час розроблення іншими мовами, таких як Ruby, Python і Perl за допомогою сполучних бібліотек (RubyCocoa, PyObjC і CamelBones відповідно). Також можна писати Cocoa-програми на Objective-C у звичайному текстовому редакторі і вручну компілювати їх за допомогою GCC або make-сценаріїв для GNUstep. Cocoa-програми зазвичай мають характерний зовнішній вигляд, оскільки це середовище багато в чому спрощує підтримку принципів «дружнього інтерфейсу» Apple (Apple Human Interface Guidelines).

Середовище Cocoa суворо дотримується архітектурного шаблону Модель – Відображення – Контролер (Model – View – Controller, MVC). В Mac OS X компанія Apple представила сімейство класів MVC, що забезпечує стандартну функціональність поведінки – NSController. Ці класи вважаються частиною системи Cocoa Bindings, яка широко використовує такі протоколи, як Key-Value Coding (KVC) та Key-Value Observing (KVO). KVC дозволяє звертатися до елемента даних або властивості об’єкта, а також змінювати його під час виконання програми за ім’ям – ім’я властивості виступає ключем до його значення. KVC приводить до надзвичайної гнучкості дизайну – тип об’єкта знати необов’язково, але будь-яка його властивість може бути отримана за допомогою KVC. Крім того, завдяки технології KVO забезпечується автоматична синхронізація властивостей об’єктів, пов’язаних між собою [5].

За рахунок використання Key-Value Observing можна легко реалізувати відслідковування змін у полях моделі програмного забезпечення, а використання Key-Value Coding надає універсальний спосіб доступу до значень. Отже, можна просто пов’язати поля моделі з візуальними компонентами інтерфейсу користувача так, що зміни в одному місці будуть автоматично передаватись у всі решта.

Отже, вибір програмної технології Cocoa для розв’язання поставленої задачі є обґрунтованим та доцільним.

Для того, щоб програма виконувала своє основне призначення, у ній було реалізовано низку алгоритмів для опрацювання зображень.

Зокрема реалізовано базові методи регулювання колірних характеристик зображення, таких як: яскравість, контрастність, насиченість, інтенсивність тіней. Базуючись на HSV моделі, пропонуємо встановлювати:

– значення насиченості – це інтенсивність кольору, що визначається як відсоток кольору вмісту певного тону відносно вмісту сірого і може змінюватися від 0 до 100 % (нульова насиченість відповідає абсолютно сірому кольору);
– значення яскравості – характеристики, що визначає, наскільки світлим чи темним може бути певний колір; цей параметр може змінюватися в діапазоні від 0 до 100 %;
– значення контрастності – максимальне відношення яскравості або щільності найсвітлішої і найтемнішої точки зображення.

Для усунення ефекту червоних очей реалізовано такий алгоритм [6]:

- розпізнавання шкіри об’єктів на ділянки зображення алгоритмом швидкої сегментації зображення;
- опрацювання ділянок зображення, на яких зображена шкіра, методами математичної морфології;
- перетворення зображення в напівтонове, виділяючи тим самим ділянки червоних очей;
- виявлення і сегментація частин зображення, на яких зображена шкіра, та присутній ефект червоних очей;
- заміна червоного кольору очей для отримання природного вигляду обличчя та очей.

Також реалізовано ефект розмиття та підвищення чіткості зображення. Розмивання зображення реалізовано методом Гаусса [7]. Цей ефект широко використовується для зменшення шуму в зображенні та зниження деталізації. Розмивання Гаусса – це тип фільтра розмивання зображення, що використовує функцію Гаусса для розрахунку трансформації кожного пікселя у зображенні. Коли метод застосовується у двох вимірах, то отримується поверхня, контури якої є концентричними колами розподілу Гаусса з центральної точки. Значення з цього розподілу використовують для створення матриці згортки. Для кожного нового значення пікселя визначається середнє зважене в околі пікселя. Значення поточного оригінального пікселя має більшу вагу (найвище значення розподілу Гаусса), а сусідні пікселі отримують все меншу вагу залежно від того наскільки далеко вони перебувають від поточного оригінального пікселя. Це надає ефект розмитості, яка зберігає межі та краї краще ніж інші, аналогічніші фільтри розмиття [7].

Програмне забезпечення спроектовано та розроблено так, щоб графічний інтерфейс користувача не перешкоджав роботі основних методів опрацювання зображення у разі його модифікацій. На рис. 1 наведено діаграму компонент розробленого програмного забезпечення.

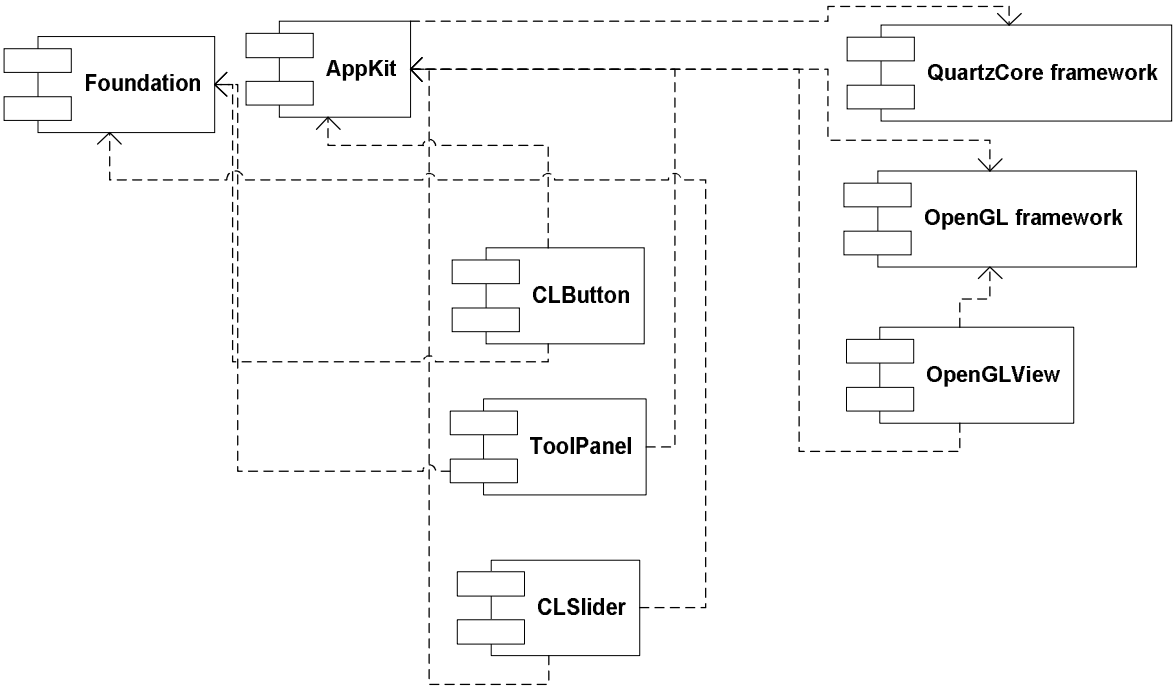

Рис. 1. Діаграма компонент програмного забезпечення

У графічному інтерфейсі користувача передбачено нестандартні візуальні компоненти, тому для кожної категорії об’єктів створено свій клас. За допомогою Interface Builder сконструйовано головне вікно програми, три елементи відображення – панелі з інструментами для редагування, ретушування зображення, інформаційної панелі; елемент відображення типу OpenGLView; панель з базовими елементами (повноекранний режим, повернути/повторити дію, вказівник).

Для реалізації розсувних панелей інструментів використано три види анімацій: анімація зміни розміру панелі, анімація руху панелі, анімація заміни однієї панелі відображення іншою. На рис. 2 наведено діаграму класів для реалізації інтерфейсу користувача.

Функціональна логіка графічного інтерфейсу програми зосереджена в трьох окремих класах-контролерах. Перший з них відповідає за управління в графічній ділянці редагування зображення. Другий клас-контролер відповідає за панель ретушування зображення. Третій клас-контролер відповідає за управління головним вікном програми. На ньому розміщується навігаційна панель управління програми з елементами управління, які відповідають за завантаження зображень, збереження зображень, зміну розмірів зображення в робочій ділянці, повернення/повтор дії, увімкнення/вимкнення повноекранного режиму.

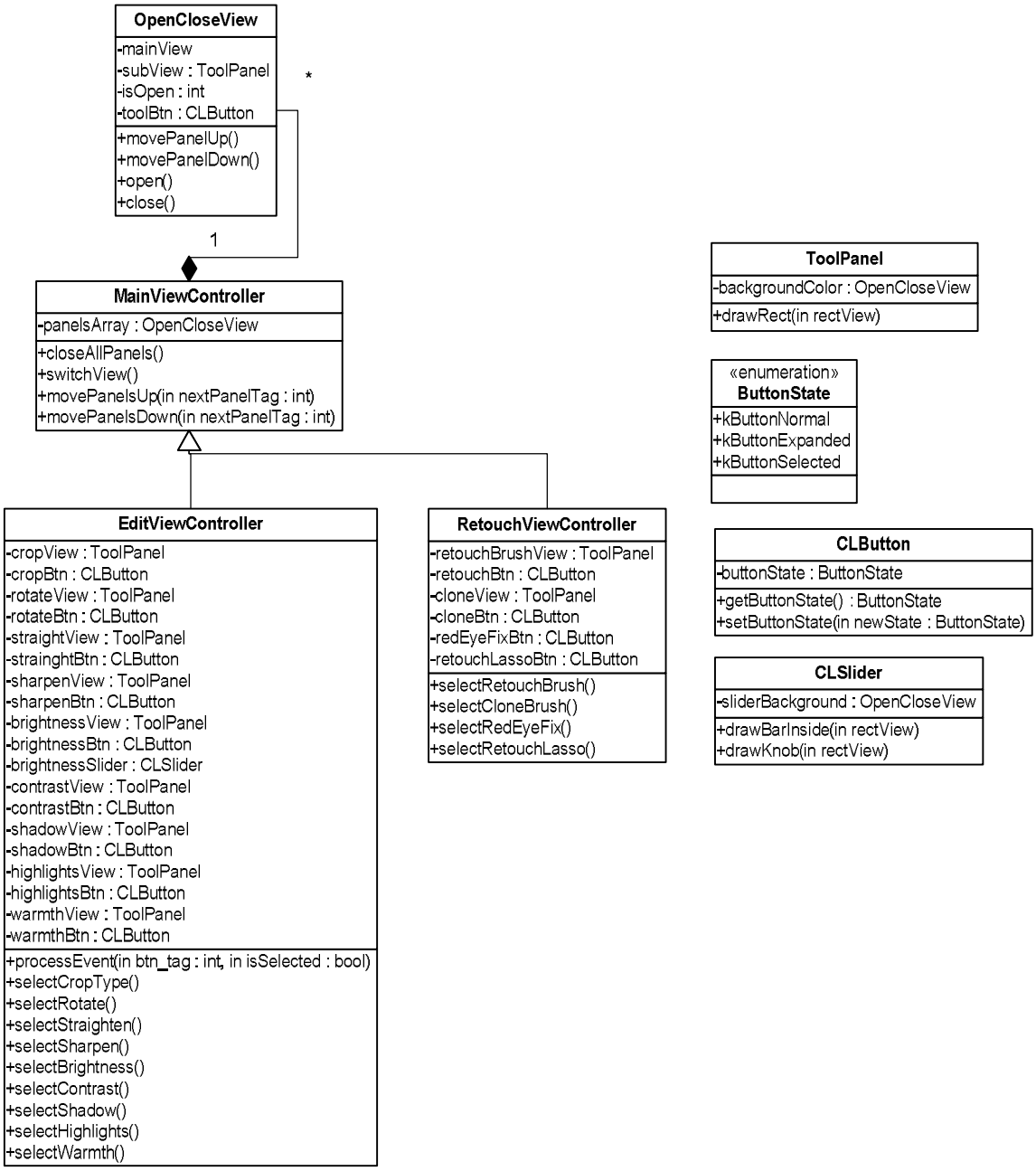

Рис. 2. Діаграма класів інтерфейсу користувача

Основою робочої ділянці зображення є панель з елементом OpenGLView. Цей клас дозволяє здійснювати маніпуляції над окремими пікселями зображення, на основі яких базуються алгоритми основних функцій програми.

У програмному продукті реалізовано логіку елемента управління TabView, який забезпечує перемикання між різними сторінками-табами навігаційної панелі, оскільки базовий елемент NSTabView не дозволяє реалізувати можливості, зокрема, встановлення зображень на кнопки-таби. Саме тому було реалізовано три окремі панелі, що завантажуються за викликом конкретної таб-кнопки, які, своєю чергою, реалізовані як звичайні кнопки.

Кожна окрема функція навігаційної панелі візуально складається з кнопки та панелі з піделементами управління. У програмі реалізована логіка відкривання, закривання панелі, зсув інших панелей догори/донизу відповідно до операцій над панеллю.

В інтерфейсі користувача надано особливого вигляду елементам прокрутки, кнопок, панелей відображення. Також реалізована зміна курсора відповідно до вибраного інструмента в конкретний момент часу.

Інтерфейс користувача розділений на логічні ділянки:

- панель з кнопками : відкрити, зберегти зображення;
- панель, що відображає назву, розміри та масштаб зображення, а також кнопки скасувати/повернути дію, вказівник, увімкнути/вимкнути повноекранний режим;
- ділянка роботи із зображення;
- ділянка інструментів для обробки зображення, представлена трьома вкладками: інструменти для ретушування зображення, редагування та інформація про зображення.

На рис. 3 зображено приклад роботи програми на вкладці редагування зображення.

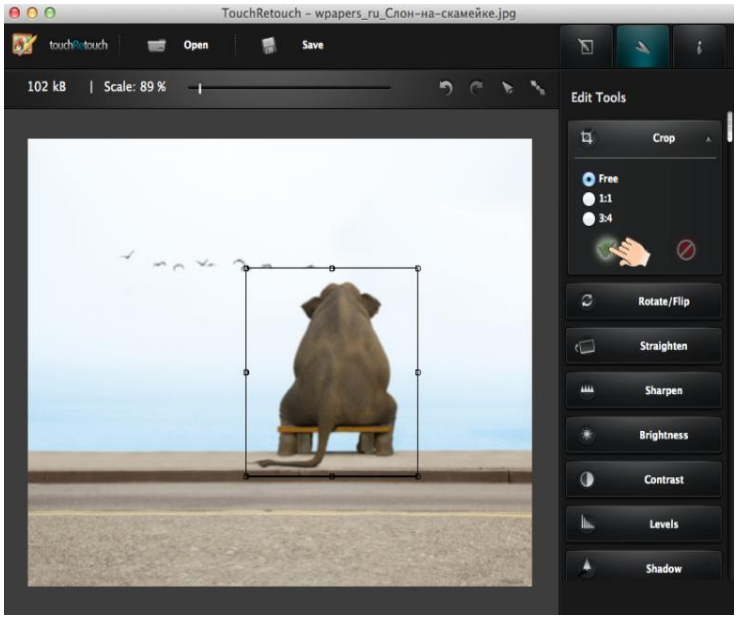

Рис. 3. Інтерфейс користувача розробленої програми

Кожна з вкладок містить різну кількість інструментів, які грамотно впорядковані і дають можливість користувачу зручно вибирати інструмент. Інтерфейс спроектований так, щоб усі решта інструментів не відволікали і не заважали користувачу, але й враховано те, що користувач може мати доступ до решти вкладок за потребою. Для цього створено панель, яка містить елементи управління конкретним ефектом. Така панель плавно відкривається і закривається за потребою користувача. На кожній з двох вкладок (ретушування, редагування) розміщено необхідну кількість таких панелей залежно від потреби.

Панель представляється класом з набором властивостей елементів управління та панеллю відображення, яка має нестандартний вигляд, тому в програмній реалізації для неї створено окремий клас. Для кожної з двох вкладок передбачено масив таких елементів, якими керує контролер вікна програми.

Одночасно користувач може працювати лише з однією вкладкою, тому передбачено автоматичне закривання відкритої панелі під час відкривання іншої. На діаграмі послідовності (рис. 4) виконання продемонстровано один з випадків – відкривання панелі, у той час як вже є відкрита інша.

В інтерфейсі також передбачено зміну вигляду курсору, оскільки користувачу повинно бути зрозуміло, яку операцію він тепер здійснює. Вигляд курсора може залежати не лише від вибраного інструмента, але від таких елементів управління, як повзунок (наприклад, зміна радіуса кисті). Для встановлення значення контрастності, чіткості, інтенсивності тіней та інших ефектів використовується компонент «повзунок», який дозволяє користувачу з легкістю встановити потрібне значення того чи іншого параметра.

Запропоновані алгоритмічно-програмні засоби перевірено на ефективність через тестування зручності використання програмного забезпечення набором з 14 окремих тестових випадків.

Акцент був зроблений на швидкості та ефективності навігації користувача по головному меню програми: панелі, кнопки, та інші елементи управління. У результаті виконання всіх тестових випадків було отримано очікувані результати.

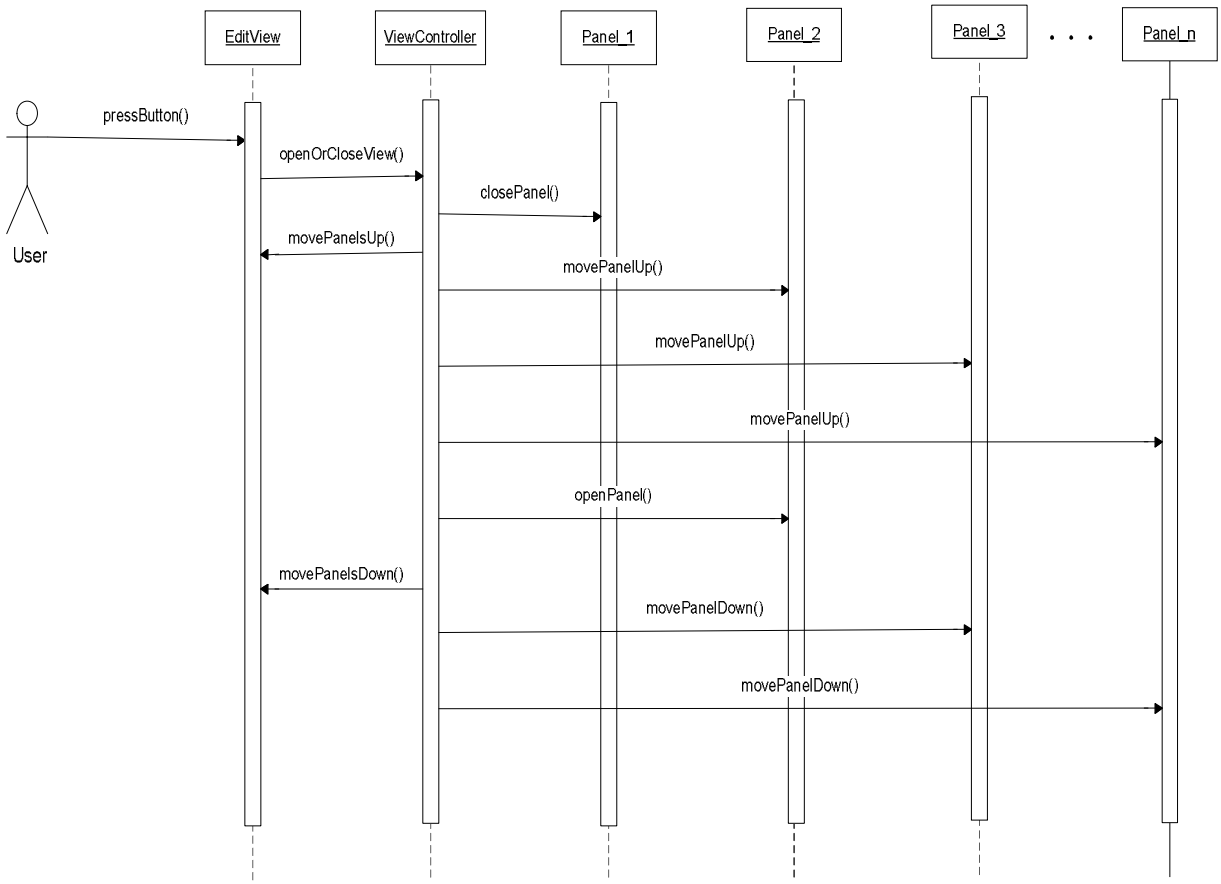

Рис. 4. Діаграма послідовності роботи користувача з нестандартними панелями

Висновки

Запропоновані алгоритмічні та програмні засоби опрацювання зображень в операційній системі Mac OS X забезпечують виконання таких операцій: видалення об'єкта; клонування об'єкта; зміну яскравості, контрастності, чіткості зображення; зміну інтенсивності тіней та інтенсивності теплих відтінків; усунення ефекту червоних очей. Результати тестування підтвердили ефективність розроблених засобів опрацювання зображень. Розроблене прогамне забезпечення орієнтоване на пересічного користувача, має привабливий графічний інтерфейс.

1. Статистика операційних систем [Електронний ресурс] – Режим доступу: http://uk.wikipedia.org/wiki/Статистика_популярності_операційних_систем. 2. Офіційний сайт компанії Apple Mac App Store [Електронний ресурс]. – Режим доступу: http://www.apple.com/mac/app-store. 3. Kernel Architecture Overview [Електронний ресурс] – Режим доступу: https://developer.apple.com/library/mac/#documentation/Darwin/Conceptual/ KernelProgramming/Architecture/Architecture.html. 4. Aaron Hillegass Cocoa Programming for Mac OS X / Aaron Hillegass, Adam Preble – 4th ed, 2011. – 400 p. 5. Cocoa Design Patterns / Erik M. Buck, Donald A. Yacktman – 1th end, 2009. – 456 pages. 6. Towards automatic redeye effect removal [Електронній ресурс] – Режім доступу : www.comm.toronto.edu/~kostas/Publications2008/pub. 7. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001.