

Разработка приложения обработки графических изображений под ОС Android

Арутюнян Мартирос Геннадьевич
Сахалинский государственный университет
Студент

Козлов Евгений Николаевич
Сахалинский государственный университет
старший преподаватель кафедры информатики

Аннотация

В статье рассматривается процесс проектирования и разработки приложения для обработки изображений под ОС Android. Представлены реализованные функции и скриншоты проекта.

Ключевые слова: обработка изображений, программный код, Android, фильтры изображений.

Development of graphic image processing application for the Android OS

Arutjunjan Martiros Gennadievich
Sakhalin State University
Student

Kozlov Evgenii Nikolaevich
Sakhalin State University
IT Department Lecturer

Abstract

In the article discusses the process of designing and developing applications for processing images for the Android OS. Presented implemented features and screenshots of the project.

Keywords: image processing, software code, Android, image filters.

В современном мире всё большее количество высокотехнологичных электронных устройств завоевывают самые разные области нашей жизни. Сегодня практически никто не представляет свою жизнь без смартфонов и планшетов. С помощью них решается множество вопросов и проблем, они спасают в критической и даже опасной для жизни ситуации. Кроме того, все больше людей отказываются от приобретения фотоаппарата в пользу покупки смартфона с хорошей камерой. Сделанные на смартфон фото удобно размещать в социальных сетях или же в файлообменниках. Тем самым с легкостью сохранить, вспоминая о чем либо, ведь в наши дни лишь

немногие печатают фотографии и хранят их в бумажном виде. Но теперь количество фотографий увеличилось в миллионы раз. В основном из-за их большого количества, сделанных на мобильных устройствах.

Существует множество программ для обработки изображений на персональных компьютерах. Но сейчас смартфоны используются чаще компьютеров, следовательно, появляется задача переноса приложений или же создания новых под мобильные платформы. Таким образом, было разработано приложение для обработки графических изображений, в котором были учтены все недостатки существующих приложений и добавлен функционал, позволяющий разнообразить графическую обработку изображений на мобильных устройствах.

Разработка мобильного приложения начинается с определения платформы, для которой оно будет разрабатываться. В настоящее время одной из наиболее популярных операционных систем для мобильных устройств является ОС Android компании Google, основанная на ядре Linux. Многие пользователи отдают предпочтение именно этой операционной системе, основывая свой выбор на функциональности и удобстве использования и доступности устройств на базе этой ОС. Учитывая наличие фотокамер у большинства смартфонов и планшетов, работающих под управлением ОС Android, а так же большую любовь к фотографии у людей, приложения обработки изображений пользуются огромной популярностью. Их количество давно перевалило за сотню, но основной ошибкой рынка до сих пор остается однообразие.

После анализа популярных сред разработки, была выбрана интегрированная среда AndroidStudio, так как она является самой популярной, находится в свободном доступе и разрабатывается компанией-разработчиком самой операционной системы Android.

В приложении реализованы как функции стандартных редакторов изображений, так и добавлены дополнительные, которые основаны на комбинации нескольких простых эффектов для создания более сложного.

Например, из стандартных были добавлены цифровые фильтры изображений:

1) «оттенок серого», вычисляемый с помощью использования ступенчатого изображение равномерного ряда оптических плотностей нейтрально-серых полей (Рисунок 1)

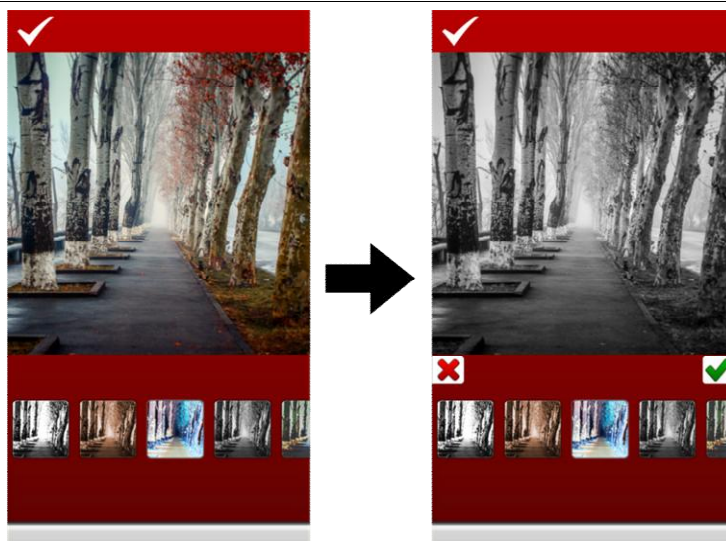


Рисунок 1 – «Оттенок серого»

2) «сепия», алгоритм получения эффекта которой схож с алгоритмом «оттенка серого», но после получения значений с каждого пикселя, для них на каждом канале определяется уровень интенсивности Сепия-тонирования. (Рисунок 2)

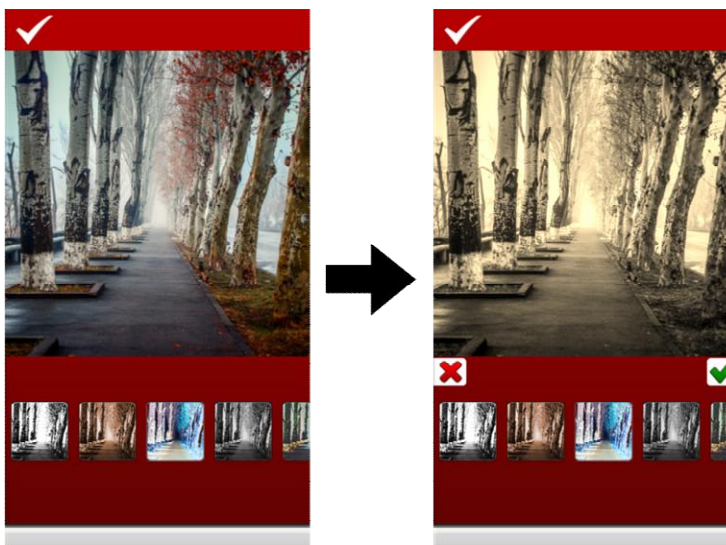


Рисунок 2 – «Сепия»

3) «негатив», получения которого изображения обрабатываются таким образом, что каждый пиксель заменяется значения всех трех RGB-каналов на его дополнение до 255. Тем самым значения на каналах инвертируются. (Рисунок 3)

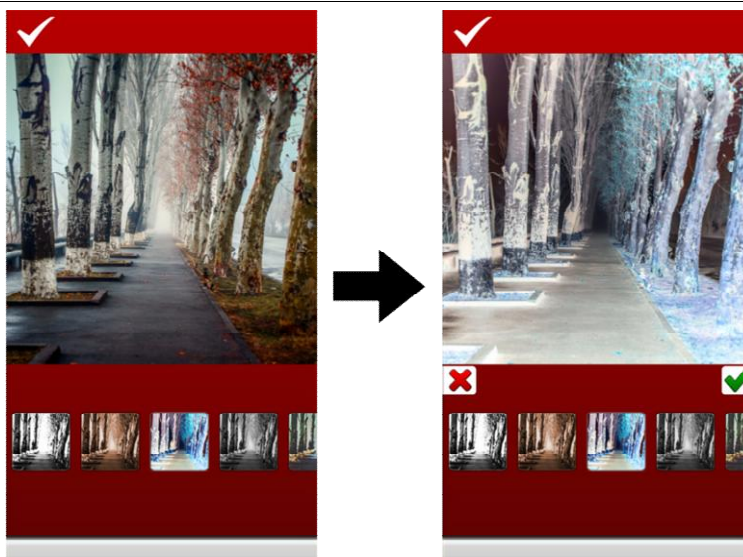


Рисунок 3 – «Негатив»

Помимо фильтров присутствуют такие эффекты как изменения яркости и контраста изображения, фильтр улучшения четкости, а также различные фильтры наложения.

На основе алгоритмов лежащих в основе этих стандартных эффектов были построены более сложные. Например, функции масштабирования легли в основу создания эффекта полароидного изображения. (Рисунок 4)

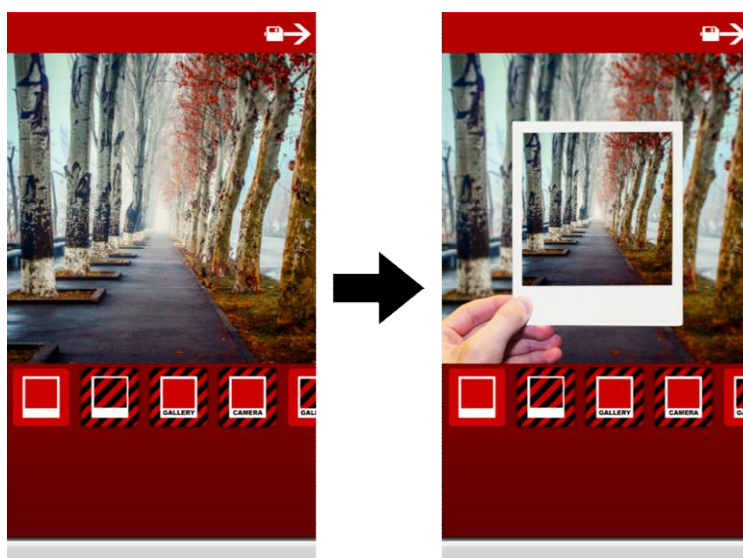


Рисунок 4 – Эффект полароида

Дальнейшее развитие разработки в направлении совмещения эффектов, позволило за счет добавления к масштабируемости цветowych фильтров создать функционал «имитации карандашного рисунка», в котором используется следующий алгоритм:

- 1) К исходному изображению применяется эффект оттенка серого и создается копия черно-белого варианта.
- 2) Инвертируются цвета полученного изображения.

3) К негативу изображения применяется фильтр размытия по Гауссу.

4) Выходным результатом будет наложение полученного после размытия изображения и черно-белой копии исходного методом Color-Dodge(Осветление основы).

Режим «Осветление основы» уменьшает контраст между совмещенным и базовым цветами, осветляя базовый цвет для отражения совмещенного цвета. Общая формула для вычисления этого эффекта выглядит следующим образом:

$$\frac{b}{255 - a} \cdot 255, \text{ где}$$

a - значения пикселей исходного изображения

b – значения пикселей наложенное изображения

В результате получается эффект карандашного рисунка наложенного на фотографию. (Рисунок 5)

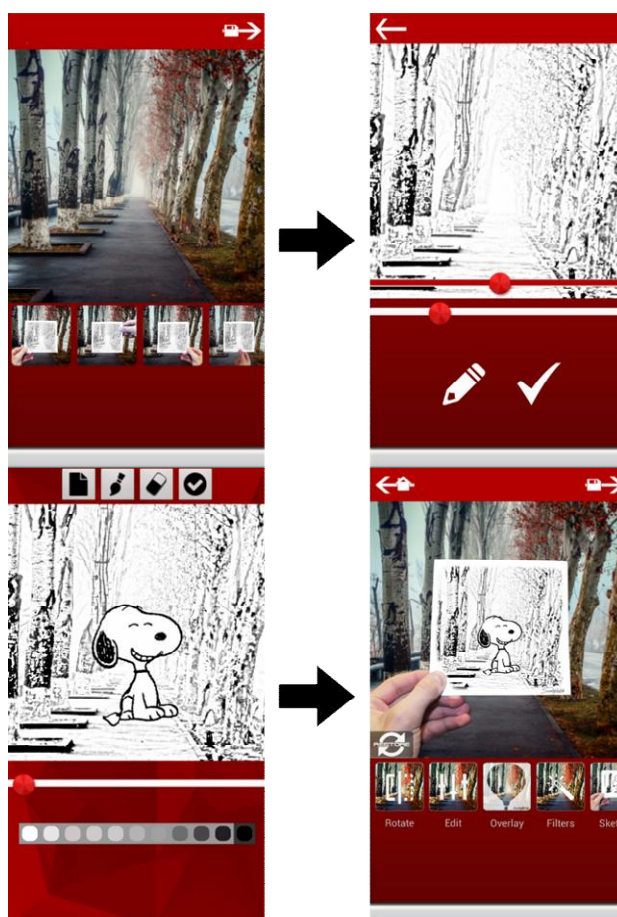


Рисунок 5 – Комбинация эффектов

Помимо функционала для пользователя в приложении была учтена возможность эффективной загрузки больших изображений. Чтобы избежать `java.lang.OutOfMemory` исключения, нужно проверяются размеры растрового изображения перед его декодированием.

Когда размеры изображения известны, они используются для того, чтобы решить, можно ли полное изображение загружать в память или нужно загрузить субдискретизированную (subsampled) версию. Вот некоторые факторы для принятия решения:

- 1) Расчетное потребление памяти загрузки полного изображения в памяти.
- 2) Объем памяти для загрузки изображения для любых других требований памяти вашего приложения.
- 3) Размеры ImageView или интерфейсного компонента, в который будет загружаться изображения.
- 4) Размер экрана устройства.

Например, не нужно загружать 1024x768 пикселей изображение в памяти, если оно в конечном итоге будет отображаться в миниатюре 128x96 пикселей в ImageView. Тем самым на выходе получается декодированное изображение, которое использует меньше места в памяти при меньшей потере качества самого изображения.

Люди стремятся к совершенству и хотят, чтобы на фотографиях важные для них события и близкие им люди выглядели идеально. Но даже лучшие фотоаппараты не могут компенсировать недостаток опыта, поэтому снимки не всегда получаются удачными, а вернуть прошедшее мгновение невозможно. В итоге, накапливаются затемненные, расплывчатые и засвеченные фотографии, возникает вопрос обработки изображений. В соответствии с данным аспектом возникает необходимость создания приложений обработки фотографий и сохранения ценных моментов в наиболее качественном виде чему и была посвящена данная работа.

В результате было разработано приложение для обработки графических изображений под операционную систему (ОС) Android. Были проведены тестирования приложения на разных устройствах, в ходе которых были выявлены и исправлены ошибки, касающиеся корректного отображения изображений на устройствах с большим разрешением экрана.

Библиографический список

1. Майер Р. Android 4. Программирование приложений для планшетных компьютеров и смартфонов. М: Эксмо, 2013. 816 с.
2. Кулешов А., Сушкова Л., Шевченко Н. Методы и алгоритмы обработки изображений. Lambert Academic Publishing, 2012. 104 с.
3. Потапов А.А., Пахомов А.А., Никитин С.А., Гуляев Ю.В. Новейшие методы обработки изображений. М.: Физматлит, 2008. 496 с.