

## РАЗРАБОТКА ГРАФИЧЕСКОГО РЕДАКТОРА ОПТИМАЛЬНОЙ ЦВЕТКОРРЕКЦИИ

**Влацкая И.В., канд. техн. наук, доцент,  
Баранов Д.А., Влацкая Е.Ф.  
Оренбургский государственный университет**

С развитием цифровой фотографии и её доступности широким слоям населения стало появляться всё больше графических редакторов, направленных на обработку изображений. Одни из них представляют собой объёмные профессиональные продукты, в функционал которых заложено большое число возможностей, способных удовлетворить потребности профессионалов на самом высоком уровне. Многофункциональность таких программ несёт в себе ряд неудобств для рядового пользователя. Несмотря на широкий выбор инструментов в профессиональных программах обработки изображений, эти функции зачастую являются избыточными. В таких программах обычно несколько инструментов, регулирующих яркость или отвечающих за коррекцию цвета, причем используя эти функции, можно добиться одного и того же результата. Интересно, что даже в продуктах одной компании Adobe Photoshop и Adobe Lightroom инструменты цветокоррекции различаются. При этом иногда появляется необходимость использовать для обработки изображения инструменты как первой, так и второй программы. Это не совсем удобно, поскольку при обработке в нескольких редакторах из-за необходимости повторного сохранения, которое влечет за собой сжатие, теряется часть качества.

Второй тип программ объединяет средние и мелкие продукты с меньшим функционалом, но проще в освоении и требованиях. Такие программы имеют более узкую направленность, что ограничивает действия пользователя, но не даёт ему запутаться. Часто они не требуют лицензии, что значительно упрощает работу. Ограниченность программы набором функций для решения всего нескольких задач может быть не только её недостатком, но и преимуществом: этим задачам уделяется больше внимания и представлены различные способы их решения. Такие программы являются оптимальным выбором для пользователя, который применяет обычно один набор функций, например, только поправляет яркость/контраст и цвет изображения.

Таким образом, сравнивая функциональные характеристики существующих графических редакторов, приходим к тому, что все программы имеют как достоинства, так и недостатки. Для создания нового графического редактора необходимо уточнить задачи, подлежащие решению, чтобы, если это возможно, избежать недостатков существующих программ. Таким образом, можно сформулировать следующие требования к разработке графического редактора:

- простота освоения. Необходимо максимально сократить время, затрачиваемое пользователем на изучение интерфейса программы и её функциональ-

ных возможностей. С этой целью можно разработать справку программы и реализовать стандартный пользовательский интерфейс;

- оптимизация под многоядерные процессоры продиктована современной техникой. В настоящий момент почти все компьютеры и смартфоны имеют несколько ядер, а изображения хранят в себе всё больше мегапикселей, что означает объёмную работу для функций графического редактора;

- наличие алгоритмов цветокоррекции, ускоряющих работу с цветом. Все графические редакторы имеют такие инструменты цветокоррекции, которые необходимо изучать и набирать опыт, прежде чем будет получаться стабильный результат. Кроме того, от пользователя необходимо понимание цвета, чтобы двигать ползунки не наугад, а точно зная, изменение каких величин принесёт за собой желаемый результат. Необходимо сохранить количество кликов пользователя, позволяющих добиться нужного результата;

- расширенные возможности выборочной коррекции цвета. Необходимо объединить различные варианты выборочной цветокоррекции, представленные в рассмотренных графических редакторах, в одной функции для наиболее полного использования их возможностей;

- построение цветовой схемы по изображению;

- обработка изображения в соответствии с цветовой схемой позволит создавать изображения, подходящие к цветовой схеме web-ресурса.

Решив поставленные задачи, можно будет устранить некоторые недостатки рассмотренных графических редакторов и повысить эффективность цветокоррекции изображений.

Существует множество средств, значительно упрощающих разработку программного средства. При выборе учитывались следующие критерии:

- свободный доступ к программному обеспечению;
- наличие необходимого для разработки функционала;
- относительная простота использования;
- совместимость с другими средствами.

В соответствии с перечисленными критериями необходимо определить инструменты разработки проекта программного средства, среду разработки для подходящего языка программирования с возможностью создания графического интерфейса, а так же выбрать технологию параллельного программирования, призванную повысить эффективность алгоритма цветокоррекции.

В основе разрабатываемого редактора лежит алгоритм статистической цветокоррекции. Целью этого способа цветокоррекции является придание исходному изображению колорита другого изображения так, чтобы исходное выглядело естественным и сохранило свои особенности. С помощью этого алгоритма можно избавиться от нежелательных оттенков на изображении или придать желаемые.

Метод заключается в том, что прежде всего вычисляются математическое ожидание и дисперсия цвета на обоих изображениях. После применения алгоритма математическое ожидание цвета целевого изображения заменяется на

математическое ожидание изображения-источника цвета. Аналогично меняется дисперсия. Для вычислений применяются формулы:

$$C_t^{new} = E_s + (C_t - E_t) \frac{D_s}{D_t}, \quad (1)$$

$$E_i = \frac{1}{n_i} \sum_{j=1}^{n_i} C_i^j, \quad (2)$$

$$D_i = \sqrt{\frac{1}{n_i} \sum_{j=1}^{n_i} (C_i^j - E_i)^2}, \quad (3)$$

$$i \in \{s, t\}, \quad (4)$$

где  $C$  - цветовой канал пиксела;

$D$  - дисперсия;

$E$  - математическое ожидание;

$new$  - индекс для обозначения нового значения цветowego канала пиксела;

$s$  - индекс принадлежности пиксела к изображению-источнику цвета;

$t$  - индекс принадлежности пиксела к целевому изображению;

$n$  - количество пикселей в изображении.

Эта формула применяется к значению каждого цветowego канала каждого пиксела целевого изображения [4].

Поскольку вычисления в данном алгоритме выполняются циклически для каждого пиксела целевого и донорного изображений, целесообразно распараллелить эти циклы для увеличения скорости обработки изображения. При этом эффективнее всего разделить итерации между потоками динамически, чтобы сбалансировать время их работы. С другой стороны, в данном алгоритме можно было бы осуществить параллелизм по данным и предоставить потокам работу с разными изображениями, однако в таком случае дисбаланс времени работы потоков оказался бы существенным. Дисбаланс был бы связан с возможной разницей размеров целевого и донорного изображений, а значит и объёма вычислений.

После изучения теории и проведения практических экспериментов с помощью данного алгоритма, можно сделать следующие выводы:

- метод действительно решает поставленную задачу, т.е. осуществляет изменение колорита целевого изображения с сохранением его особенностей и результат перекраски при разумном подборе пары изображений выглядит естественно, т.е. обеспечивается ожидаемый эффект перекраски. Алгоритм также позволяет убирать нежелательные оттенки с фотографий;

- скорость работы алгоритма напрямую зависит от размера входных изображений, т.к. преобразовывается каждый пиксел. При этом скорость работы алгоритма может быть существенно повышена за счет применения технологии параллельного программирования;

- благодаря использованию статистики метод применим независимо от

размеров входных изображений и соответствия размеров между собой. В чистом виде метод перекрашивает изображение целиком;

- важным нюансом является то, что изображения должны быть схожи в своей композиции и содержании, иначе результат может быть непредсказуемым.

Главное окно программы содержит в себе область прокрутки и 3 пункта меню, соответствующие выделенным модулям.

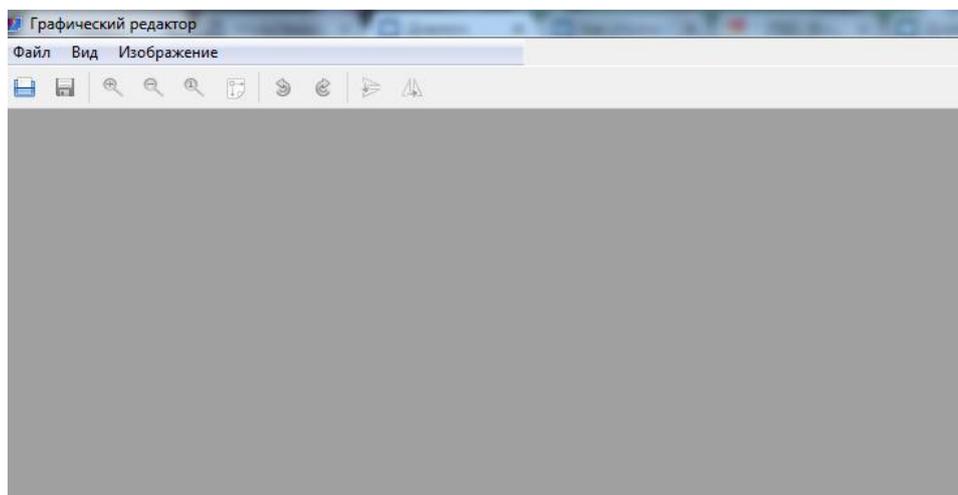


Рисунок 1- Главное окно программы цветокоррекции

После выбора пункта меню «открыть», изображение отображается по центру прокручиваемой области, подстраиваясь под размеры окна.

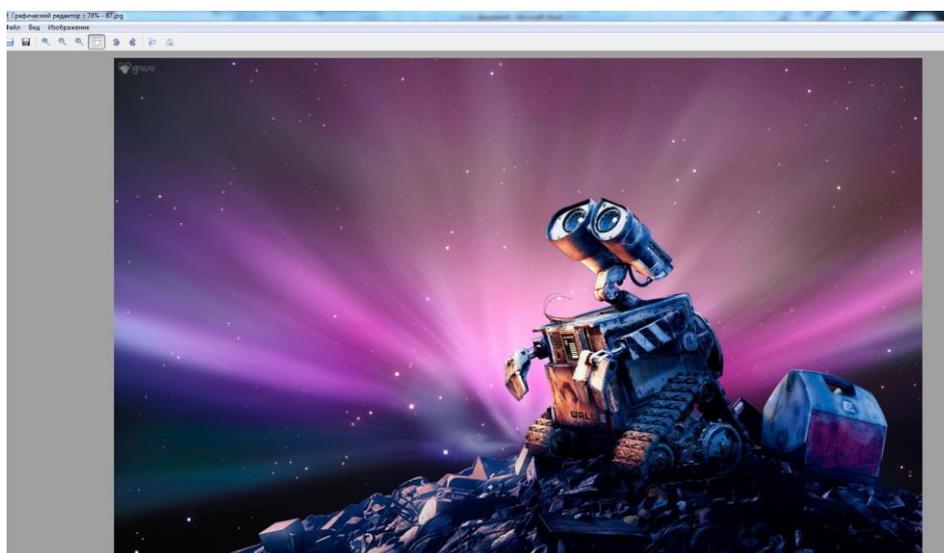


Рисунок 2- Загрузка изображения

При выборе пункта «цветокоррекция» пользователю предлагается выбрать изображение-донор.

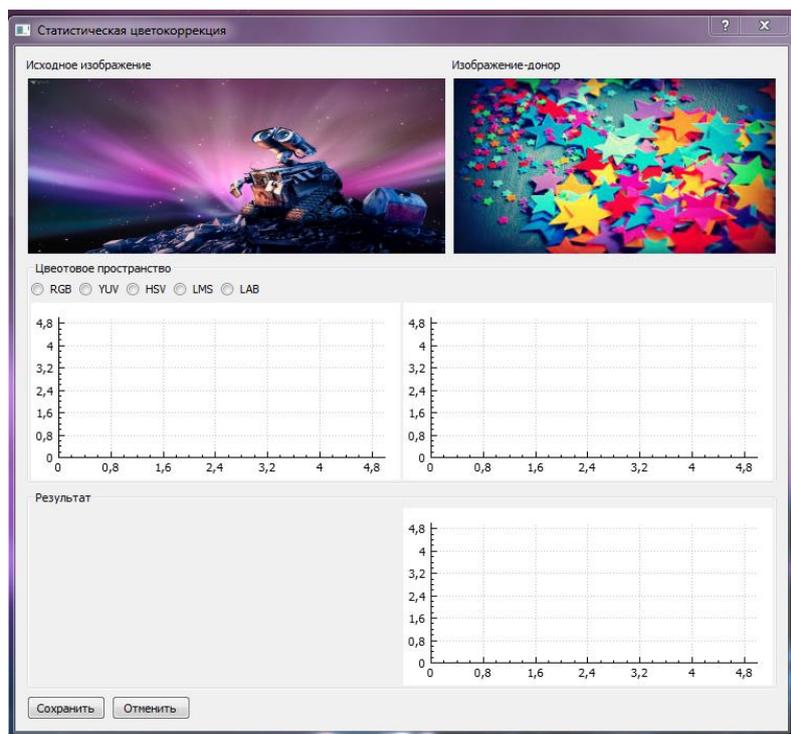


Рисунок 3 – Цветокоррекция: исходное изображение и изображение-донор.

После выбора двух изображений, пользователь может задать необходимую цветовую схему. Наиболее популярными на сегодняшний день являются следующие цветовые модели: RGB (используется в основном в мониторах и камерах), CMY(K) (используется в полиграфии), HSI (широко используется в машинном зрении и дизайне). Субтрактивная модель CMY (от англ. cyan — голубой, magenta — пурпурный, yellow — жёлтый) используется для получения твёрдых копий (печати) изображений, и в некотором роде является антиподом цветового RGB-куба. С целью унификации была разработана международная стандартная цветовая модель CIE XYZ. Модель CIE XYZ, хоть и наиболее близка к восприятию человека, но достаточно сложна в описании. В связи с этим, наибольшее распространение получило цветовое пространство lab и его модификации, представляющее все видимые цвета и оттенки в виде шара с осями L, a и b. При этом по оси L измеряется светлота (в диапазоне от 0 до 100%), отображая коэффициент спектрального отражения, по оси a измеряется красный-зеленый оттенок, по оси b оттенок желтый-синий (в диапазонах от -120 до +120).

Алгоритм статистической цветокоррекции предполагает 4 вычисления для каждого цветового канала каждого пиксела двух изображений. При этом размер изображения может быть любым. Современные цифровые камеры имеют большое разрешение, что влечет за собой большое количество пикселей, большой размер изображения и, соответственно, много работы для алгоритма. Для проведения исследования были выбраны два изображения с одинаковым размером 2500x3750, что является средним размеров качественной фотографии.

Таким образом, необходимо обработать 2 изображения, в каждом из которых содержится 9.375.000 пикселей.

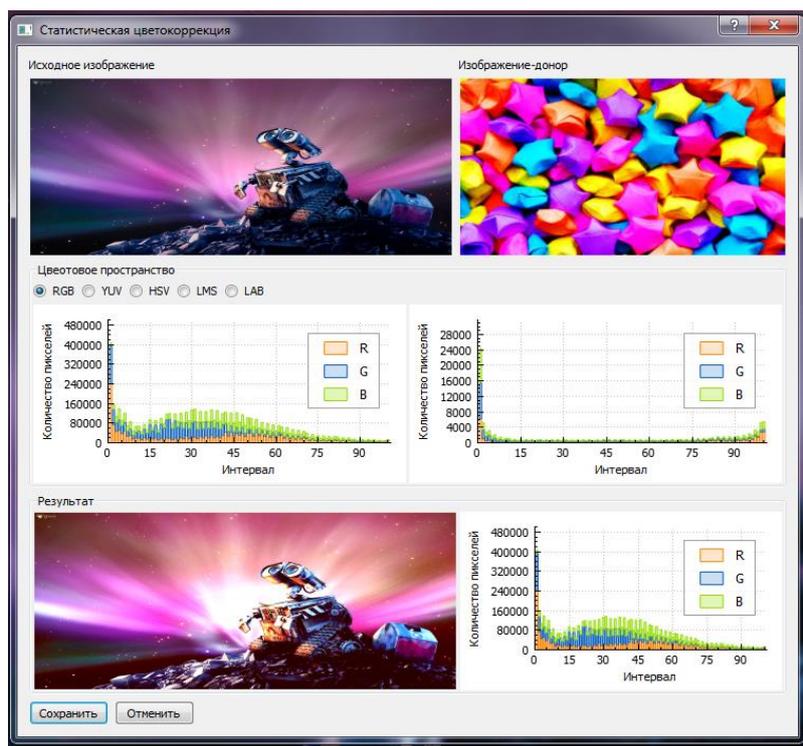


Рисунок 4 – Результат работы цветокоррекции.

Таким образом, был разработан пользовательский интерфейс, удовлетворяющий основным требованиям комфорта пользователя. В графическом интерфейсе используются стандартные элементы интерфейса, привычные пользователю, что позволяет сократить время на освоение программы. Предупреждающие сообщения снижают вероятность допущения ошибки или случайного несохранения внесенных изменений.

### 1 Список литературы

- 2 Грабалов, П.К. Компьютерная графика и основные графические редакторы / П.К. Грабалов – Калининград.: Аст, 2003 – 207 с.
- 3 Яхонтов, В.Н. Компьютерная графика / В.Н. Яхонтов – М.: ТИСБИ, 2003. – 320 с.
- 4 Косенко, П. Живая цифра / П. Косенко – М.: Тримедиа, 2013. – 286 с.
- 5 Иттен, Й. Искусство цвета / Й. Иттен – М.: Аст, 2001. – 95 с.
- 6 Крашенинников, В. Р. Основы теории обработки изображений / В.Р. Крашенинников – Ульяновск: УлГТУ, 2003. – 150 с.
- 7 Прэтт У. Цифровая обработка изображений / У. Прэтт – М.: Мир, 1982. Т. 1, 2. 791 с.

8 Фисенко, В.Т. Компьютерная обработка и распознавание изображений / В.Т. Фисенко, Т.Ю. Фисенко – СПб: СПбГУ ИТМО, 2008. – 192 с.