

УДК 004.932

Д.С. Химич, А.Ю. ХаритоновДонецкий национальный технический университет, г. Донецк
Кафедра компьютерных систем мониторинга**ТЕХНОЛОГИИ ОПРЕДЕЛЕНИЯ ДОМИНИРУЮЩИХ ЦВЕТОВ НА
РАСТРОВОМ ИЗОБРАЖЕНИИ****Аннотация**

Химич Д.С., Харитонов А.Ю. Определение доминирующих цветов на растровом изображении. При проектировании веб-сайтов, графических интерфейсов для программных продуктов и т.д. дизайнеру необходимо тщательно подойти к выбору цветовой схемы. Правильное сочетание цветов или их гармония – это уверенность в том, что пользователь отдаст предпочтение именно Вашей работе среди многих других. В данной статье рассматриваются технологии, позволяющие реализовать программную систему для определения доминирующих цветов на растровом изображении.

Ключевые слова: *дизайн, веб-сайт, графический интерфейс, растровое изображение, цветовая схема, технологии.*

Постановка проблемы. Цвета являются чрезвычайно мощным фактором воздействия на пользователя: они могут усиливать правильный тон и передавать необходимые эмоции пользователям, могут взволновать, вызвать множество чувств и стимулировать к действиям. Иногда идеальное сочетание цветов можно наблюдать в окружающем нас мире: кружка кофе на столе, ваза с цветами и т.д. Такую палитру разумно использовать в тематических разработках, например, цветовая палитра красных роз в вазе может быть взята за основу сайта по доставке живых цветов. При выборе цветовой гаммы для сайта или приложения важно сделать это правильно, руководствуясь основными принципами теории цвета. Для получения такой палитры существует большое количество программных средств, разработанные как для сетевого использования (веб-сайты), так и для локального (приложения для ПК и под мобильные платформы). Основная проблема статьи – выбор методов, реализация которых является наиболее простой для реализации вычисления доминирующих цветов на изображении. Актуальность данной статьи состоит в том, что многие существующие продукты, которые составляют цветовую схему из загруженного пользователем растрового изображения, дают сбои и вместо необходимого результата выдают ошибочный. В ходе исследования технологий, которые используются для программирования подобных программных систем, будет сформирована таблица со сравнительными характеристиками известных технологий, что позволит выделить наиболее качественный и удобный пользователю ресурс.

Цель статьи – выполнить анализ существующих технологий по определению доминирующих цветов на растровом изображении. По результатам таблицы определить технологию, которая является наиболее простой для реализации подобной программной системы и привести пример существующей системы, реализованной по данной технологии.

Результаты исследования. При разработке дизайна графических интерфейсов, сайтов и приложений основной задачей дизайнера является подбор цветовой палитры, которая играет ключевую роль в визуальном восприятии разрабатываемого продукта пользователем. Цвет в дизайне очень субъективен. То, что вызывает некую реакцию у одного человека, может вызвать совершенно другую реакцию у другого. Иногда это зависит от личных предпочтений, в других случаях – от культурного наследия. Теория цвета сама по себе является наукой. На изучении того, как цвета воздействуют на людей, индивидуально или в группе, некоторые делают карьеру. И здесь действительно многое можно изучать. Простое изменение определенного оттенка или цветовой насыщенности может вызвать абсолютное другое ощущение. Культурные различия означают, что то, что будет радостным и вызывать душевный подъем в одной стране, может быть воспринято как угнетающее в другой. В настоящее время стали популярны ресурсы по созданию цветowych палитр, которые формируются из растрового изображения. Оно может быть загружено на сервер веб-сайта, в программный продукт, после чего по определенным алгоритмам определяются наиболее часто повторяющиеся цвета, которые и выводятся в результате работы системы в виде цветовой палитры (набора цветов). Данную задачу можно реализовать практически на любом языке программирования, используя алгоритмы обработки изображений. В ходе исследования будет рассмотрен основной метод определения доминирующих цветов на растровом изображении – метод k-средних, который можно реализовать на языке C++ с помощью библиотеки OpenCV. Также для выполнения данной задачи может быть использован скриптовый язык разработки веб-страниц PHP.

1. Метод k-means (k-средних) – наиболее популярный метод кластеризации (распределение объектов на кластеры). Суть кластеризации состоит в том, что все исходные объекты (в данном случае пиксели) разбиваются на несколько не пересекающихся групп таким образом, чтобы объекты, попавшие в одну группу, имели сходные характеристики, в то время как у объектов из разных групп эти характеристики должны значительно отличаться. Полученные группы называются кластерами. Исходными значениями в простейшем способе для кластеризации являются координаты пикселя (x, y) , в более сложных случаях, например для полутоновых изображений, используется трехмерный вектор $(x, y, I(x, y))$, где $I(x, y)$ – градации серого и пятимерный вектор, если используется RGB. Метод k-средних – наиболее популярный метод кластеризации. Алгоритму широко отдается предпочтение из-за его простоты реализации, большой скорости.

Действие алгоритма таково, что он стремится минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров (центроидов):

$$V = \sum_{i=0}^k \sum_{x_j \in S_i} (x_j - \mu_i)^2, \quad (1)$$

где k – число кластеров, S_i – полученные кластеры, $i = 1, 2, \dots, k$ и μ_i – центры масс векторов $x_j \in S_i$. Это итеративный алгоритм, который делит данное множество пикселей на k кластеров точки, которые являются максимально приближенными к их центрам, а сама кластеризация происходит за счет смещения этих центров. Принцип кластеризации методом k -средних:

– выбор из множества k пикселей пикселей, которые будут центроидами соответствующих k кластеров;

– выборка начальных центроидов может быть как случайной, так и по определенному алгоритму;

– вход в цикл, который продолжается до тех пор, пока центроиды кластеров не перестанут изменять свое положение;

– проверка каждого пикселя на выявление, к какому центроиду какого кластера он является ближайшим;

– привязка пикселя к кластеру этого центроида;

– поиск новых координат центроидов k кластеров;

– проверка координат новых центроидов. Если они соответственно равны предыдущим центроидам – выходим из цикла, если нет – возвращаемся к п. 3.

Действие алгоритма в двумерном случае представлено на рисунках 1-4. На рисунке 1 изображены исходные точки и случайно выбранные начальные точки. На рисунке 2 изображены точки, отнесенные к начальным центрам. Разбиение на плоскости – диаграмма Вороного относительно начальных центров. На рисунке 3 выполняется вычисление новых центров кластеров (поиск центра масс). Предыдущие шаги повторяются, пока алгоритм не сойдется (рис. 4).

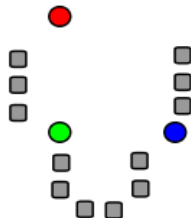


Рисунок 1 – Первый этап алгоритма k -средних

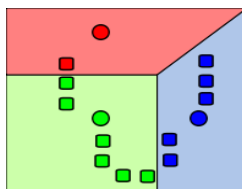


Рисунок 2 – Второй этап алгоритма k-средних

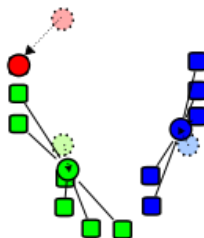


Рисунок 3 – Третий этап алгоритма k-средних

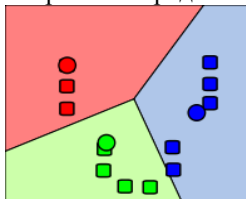


Рисунок 4 – Результат алгоритма k-средних

Однако, данный метод не лишен проблем реализации. Например, не гарантируется достижение глобального минимума суммарного квадратичного отклонения V , а только одного из локальных минимумов. Результат зависит от выбора исходных центров кластеров, их оптимальный выбор неизвестен. Число кластеров надо знать заранее.

2. OpenCV (OpenComputerVision) – библиотека компьютерного зрения с открытым исходным кодом, предоставляющая набор типов данных и численных алгоритмов для обработки изображений алгоритмами компьютерного зрения. OpenCV написана на языке высокого уровня (C/C++) и содержит алгоритмы для: интерпретации изображений, калибровки камеры по эталону, устранение оптических искажений, определение сходства, анализ перемещения объекта, определение формы объекта и слежение за объектом, 3D-реконструкция, сегментация объекта, распознавание жестов и т.д. Эта библиотека очень популярна за счёт своей открытости и возможности бесплатно использовать как в учебных, так и коммерческих целях. Модуль обработки изображений и компьютерного зрения CV позволяет производить базовые операции над изображениями (фильтрация, геометрические преобразования, преобразование цветových пространств и т.д.), анализ изображений (выбор отличительных признаков, морфология, поиск контуров,

гистограммы), анализ движения, слежение за объектами и их обнаружение, калибровка камер, элементы восстановления пространственной структуры.

3. Существует немало количество готовых PHP-скриптов для определения доминирующих цветов на изображении. Большинство из них основано на принципе поиска заданного количества наиболее повторяющихся цветов на изображении. Главной функцией в алгоритме является `GeneratorImageColorPalette::getImageColor($imageFile_URL,$numColors,$image_granularity=5)`, которая имеет три входных параметра: адрес изображения, количество цветов на выходе и шаг сетки, по которой производится статистика. Третий параметр наиболее важен – он напрямую влияет на скорость работы скрипта. Чем больше его величина, тем быстрее будет работать скрипт. По умолчанию он равен 5, но реально в изображении цвет меняется не так часто и можно поставить цифру больше значения по умолчанию. Скрипт возвращает результирующий массив в порядке возрастания цвета и количество точек, встреченных с данным цветом (`return array_slice($colors, 0, $numColors,true)`). Применение скрипта может быть самым разнообразным. У Yandex и Google реализован поиск картинок по доминирующему цвету. Другое несомненное применение данного скрипта – это сравнение двух фотографий на их схожесть, ведь скрипт дает своеобразный хеш фотографии. Сравнив два хеша и допустив небольшую погрешность, можно судить, что две фотографии содержат одни и те же данные. При этом фотографии разного разрешения, но с одинаковым содержанием также легко выявляются.

Ниже представлена таблица со сравнительными характеристиками существующих технологий по определению доминирующих цветов на растровом изображении. По результатам таблицы будет определена технология, которая является наиболее простой для реализации подобной программной системы и будет приведен пример существующего ресурса по определению доминирующих цветов, реализованной по этой технологии.

Таблица 1 – Сравнительные характеристики технологий

Характеристика	OpenCV (локальные ресурсы)	PHP (сетевые ресурсы)
Сложность реализации	+	-
Юзабилити продукта	+/-	+

Анализ данной таблицы показал, что наиболее простой в реализации и удобной в использовании программной системой является программа, разработанная на скриптовом языке PHP. В понятие юзабилити продукта входят: быстрый доступ к ресурсу (при наличии интернет-соединения), удобство пользования (ресурс не нужно скачивать и устанавливать на ПК).

Примером существующего ресурса, использующий технологию PHP, является `AdobeKuler (kuler.adobe.com)`. Загрузим растровое изображение, щелкнув на соответствующей пиктограмме (см. рис. 5).

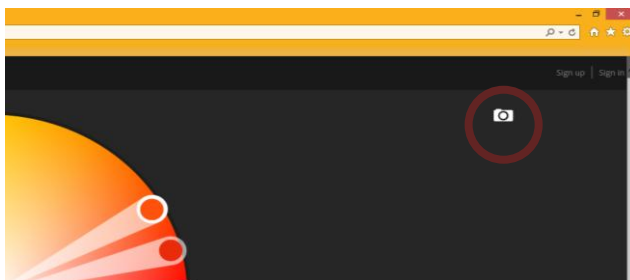


Рисунок 5 – Кнопка для загрузки растрового изображения

После обработки изображения, появляется цветовая схема из пяти доминирующих цветов (см. рис. 6).

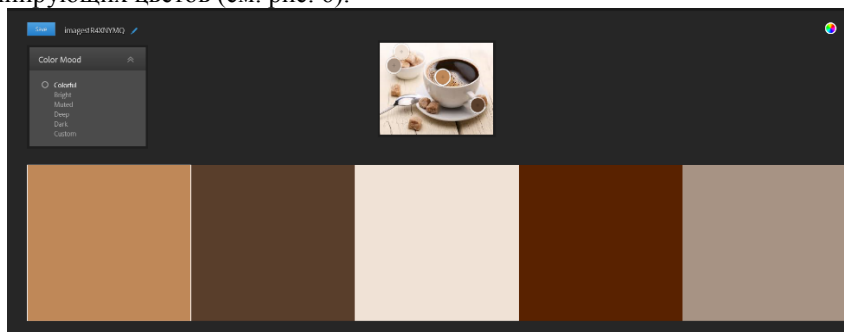


Рисунок 6 – Цветовая схема загруженного изображения

Выводы. В данной статье была проведена характеристика технологий разработки программной системы по определению доминирующих цветов на растровом изображении. Оптимальной технологией является PHP-скрипт, основанный на переборе всех пикселей изображения и группировке их в отдельные классы в порядке возрастания численности пикселей определенного цвета. Результаты данной работы могут быть использованы в личных целях пользователя, который непосредственно связан с разработкой веб-сайтов, дизайном пользовательских интерфейсов и т.д.

Список литературы

1. AdobeKuler/ Интернет-ресурс. – Режимдоступа: [www/ URL: http://www.kuler.adobe.com/](http://www.kuler.adobe.com/)
2. OpenCV/ Интернет-ресурс. – Режим доступа: [www/ URL: http://opencv.org](http://opencv.org)
3. Определение доминирующих цветов: метод k-средних/ Интернет-ресурс. – Режим доступа: [www/ URL: habrahabr.ru/post/156045/&post=18757303_424/](http://habrahabr.ru/post/156045/&post=18757303_424/)