

## АЛГОРИТМ ПОИСКА ИЗОБРАЖЕНИЙ НА ОСНОВЕ ГИСТОГРАММ

**Валюх А.А., Тонкошкур Ю.О.**

Факультет физики электроники и компьютерных систем  
Днепропетровского национального университета им. О. Гончара  
E-mail: [valiuh@gmail.com](mailto:valiuh@gmail.com), [tonkoshkur@gmail.com](mailto:tonkoshkur@gmail.com)

### **Аннотация:**

**Валюх А.А., Тонкошкур Ю.О.** *Алгоритм поиска изображений на основе гистограмм.* Рассматривается задача построения алгоритма поиска изображений по содержанию на основе использования гистограмм для каналов R, G, B. Алгоритм базируется на построении гистограмм функций распределения по всем r, g, и b компонент исходного изображения и их сравнения с гистограммами изображений среди которых могут находиться копии исходного изображения. Алгоритм защищен авторским свидетельством.

### **Общая постановка проблемы.**

Поиск изображений по содержанию — раздел [компьютерного зрения](#), решающий задачу поиска изображений, которые имеют требуемое содержание, в большом наборе цифровых изображений[1]. В качестве примера рассмотрим один из самых популярных Интернет сервисов посвященный поиску изображений по содержанию – TinEye ([tineye.com](http://tineye.com)). TinEye мгновенно анализирует исходное изображение, чтобы создать компактную цифровую подпись или «отпечаток пальца» для него[3]. TinEye ищет изображение в сети Интернет путем сравнения его отпечатков пальцев с другими проиндексированными TinEye изображениями. В данной работе реализован и протестирован алгоритм идентификации всех копий картинке по ее содержанию на основе использования гистограмм.

### **Постановка задачи.**

С целью поиска всех копий изображения целесообразно использовать методы позволяющие выполнить хеширование исходного изображения с последующим сравнением полученного хэша, с хэшами других изображений. Однако поскольку, похожие друг на друга изображения могут несколько отличаться в первую очередь по цветовой гамме, а также по ряду других показателей, то применение для этих целей таких средства как преобразование хэширующими алгоритмами типа md5 и им подобным будет не эффективно. Это очевидно, так как похожие «на глаз» изображения, но отличающиеся по своим показателям, будут давать разные хеши и, следовательно, система будет обладать низкой обобщающей способностью. Необходим алгоритм построения хэша обладающего более высокой обобщающей способностью.

Данная работа посвящена алгоритм поиска всех копий картинке по ее содержанию основанный на гистограммных хэшах.

В процессе реализации и тестирования алгоритма H AIS использовался язык Python (версия Python - 2.7) с модулями PIL, numpy, scipy и matplotlib для обработки изображения и визуализации работы алгоритма. Указанные компоненты были развернуты на системе следующей конфигурации Athlon XP 1800+ (1,53GHz), 1GB DDR OC Fedora 14

### **Основная часть.**

Рассмотрим более детально алгоритм поиска всех копий картинке по ее содержанию основанный на гистограммах. В общем виде алгоритм сводится, во-первых, к построению гистограмм распределения цветов по осям x и y по каждой из r, g и b компонент в пространстве цветов RGB. Покажем, что такой метод может быть не чувствителен к изменению масштаба исходного изображения в широком диапазоне значений (от 2 до 64 раз), а также к повороту исходного изображения на определённый угол и зашумлению как случайным RGB шумом, так и посторонними изображениями (белыми и черными линиями).

Рассмотрим суть гистограмм используемых в алгоритме. По сути гистограммы являются нормированными (приведенными к единичному масштабу) функциями, распределения цветов по осям x и y по каждой из r, g и b компонент в пространстве цветов RGB. Рассмотрим вид таких функций представленных на рис. 2 для тестового изображения на рис. 1.

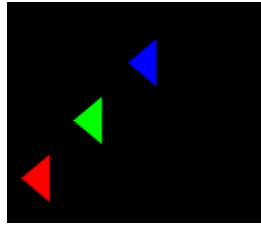


Рис.1. Тестовое изображение для демонстрации сути гистограмм алгоритма.

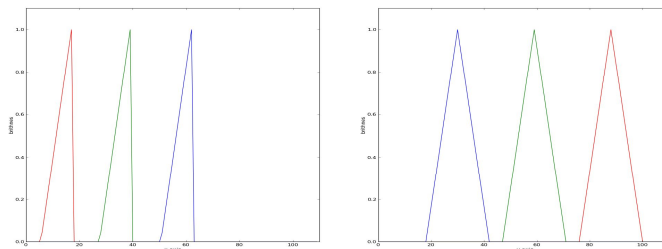


Рис.2. Гистограммы, используемые в алгоритме полученные по тестовому изображению на рис.1.

Рассмотрим более детально графики гистограмм представленных на рис. 2. В левой части изображения на рис.2 представлены три графика функции распределения по каждой из r, g и b компонент в пространстве цветов RGB по оси x изображения на рис.1[2]. Масштаб по оси x для этих графиков полностью соответствует масштабу по оси x для изображения на рис.1. Рассмотрим, к примеру, график функции распределения для r компоненты по оси x исходного изображения на рис.1. Значения этого графика  $y(x(i))$  показывает общую сумму всех значений r компоненты ( $r \in [0..255]$ ) по всей оси y исходного изображения в позиции  $x(i)$  для шкалы x исходного изображения.

$$y_r(x(i)) = \sum_{y=0}^{y=image\_height} r(x(i), y); i \in (0..image\_width); \quad (1)$$

Аналогично определяются графики для компонент g и b по оси x.

$$y_g(x(i)) = \sum_{y=0}^{y=image\_height} g(x(i), y); i \in (0..image\_width); \quad (2)$$

$$y_b(x(i)) = \sum_{y=0}^{y=image\_height} b(x(i), y); i \in (0..image\_width); \quad (3)$$

В правой части изображения на рис.2 также представлены три графика функции распределения по каждой из r, g и b компонент в пространстве цветов RGB, но по оси y изображения. Физический смысл графиков аналогичен тем, которые описаны выше. Разница состоит лишь в том, что они рассчитаны для оси y, а не x.

$$x_r(y(i)) = \sum_{x=0}^{x=image\_width} r(x, y(i)); i \in (0..image\_height); \quad (4)$$

Аналогично определяются графики для компонент g и b по оси y.

$$x_g(y(i)) = \sum_{x=0}^{x=image\_width} g(x, y(i)); i \in (0..image\_height); \quad (5)$$

$$x_b(y(i)) = \sum_{x=0}^{x=image\_width} b(x, y(i)); i \in (0..image\_height); \quad (6)$$

После расчета гистограмм их значения нормируются — то есть приводятся к единичному масштабу путем деления на максимальный элемент для каждой гистограммы.

После более детального рассмотрения сути используемых гистограмм рассмотрим этапы работы алгоритма основанного на этих гистограммах. На первом этапе работы алгоритма входное изображение приводится к единому масштабу 32x32 пикселя. Данная операция позволяет существенно сократить объем вычисления значений гистограммы для исходного изображения. Экспериментальным путем было получено, что уменьшение разрешения до 32x32 пикселя практически не влияет на общую форму гистограмм изображения, в то время как общая производительность алгоритма возрастает. Уменьшение разрешения более чем 32x32 начинает влиять на общую форму гистограмм изображения, поэтому такое изменение не рекомендуется.

После того как получена уменьшенная версия изображения, она подвергается обработке медианным фильтром. Данная операция позволяет удалить из изображения высокочастотный шум, не размывая при этом границ перехода на различных участках изображения. Данная операция позволяет сгладить гистограммы исходного изображения, убирая из них статистические выбросы, вызванные как шумом самого изображения, так и чрезмерное влияние не существенных в целом мелких деталей (пестрота изображения).

После получения всех необходимых гистограмм исходного изображения, начинается этап их сравнения с гистограммами изображений среди которых могут находиться копии исходного изображения. Сравнение происходит по всем функциям распределения  $r$ ,  $g$ , и  $b$  компонент для  $x$  и  $y$  шкал. Лишь при совпадении всех шести функций распределения в каждой паре. Под парой подразумеваются две группа гистограмм. Первая группа состоит из 3-х гистограмм для  $x$  шкалы исходного изображения и 3-х гистограмм для  $x$  шкалы проверяемого изображения. В свою очередь, вторая группа состоит из 3-х гистограмм. Такой подход выбран для увеличения «помехозащищенности» по отношению к изображениям заведомо не похожим на исходное. Поскольку вероятность одновременного случайного совпадения гистограмм по всем компонентам  $r$ ,  $g$ , и  $b$  на всех гистограммах у двух разных изображений значительно ниже чем, к примеру, вероятность одновременного случайного совпадения гистограмм по одной компоненте (либо по суммарной яркости).

Помимо прямого соответствия всех шести функций распределения ищется также соответствие между тремя функциями распределения по оси  $x$  входного изображения и тремя «зеркальными» функциями по оси  $x$  и оси  $y$  сравниваемого изображения. Аналогично ищется также соответствие между тремя функциями распределения по оси  $y$  входного изображения и тремя «зеркальными» функциями по оси  $y$  и оси  $x$  сравниваемого изображения. Под «зеркальными» функциями подразумеваются исходные функции распределения, но симметричные относительно оси  $y$ . На пример для гистограммы [1,2,3,4] «зеркальным» вариантом будет [4,3,2,1] соответственно. Операция сравнения гистограмм исходного изображения с «зеркальными» гистограммами сравниваемого изображения проводится для обеспечения инвариантности к поворотам на 90 и 180 градусов. В этом и заключается суть алгоритма или алгоритма поиска изображений на основе гистограмм.

Для тестирования работы алгоритма была создана специальная база данных изображений с использованием MySQL. В базу данных было записано описание для 1024 изображений и их гистограммы.

В процессе тестирования алгоритма было показано, что он не чувствителен к изменению масштаба исходного изображения в широком диапазоне значений (от 2 до 64 раз), линейному искажению размеров исходного изображения в широком диапазоне, а так же к поворотам на углы 90 и 180 градусов как это показано на рис.3



Рис.3. Изображения, подаваемые на вход алгоритма для проверки инвариантности к углам поворота на 90 и 180 градусам.

Также наблюдалась инвариантность к шумовым искажениям. Пример шумового искажения используемого для тестирования представлен на рис. 4б.

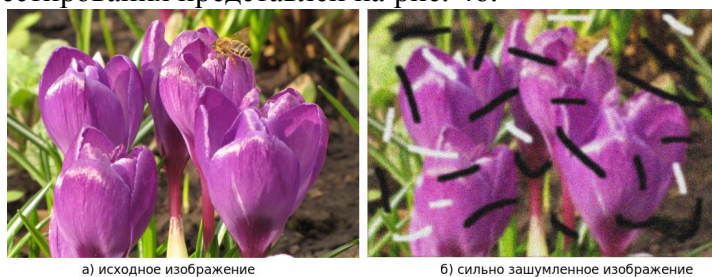
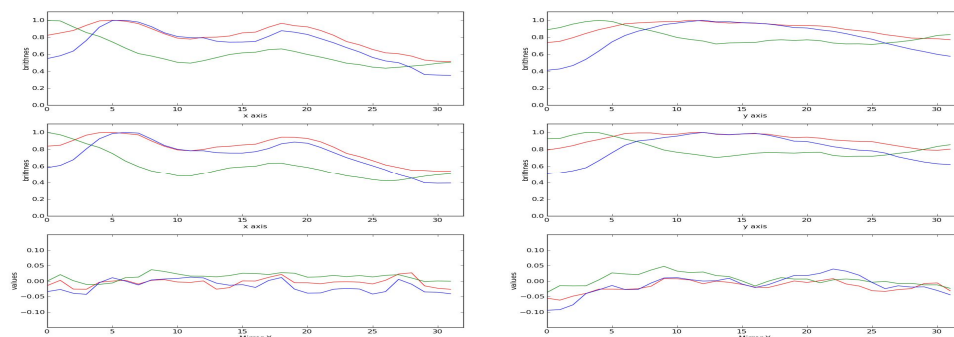


Рис.4. Изображение, подаваемое на вход алгоритма.

Изображение представленное на рис.4б было получено из изображения на рис.4а путем его зашумления случайными кривыми черного и белого цветов с последующим зашумлением «шумом RGB» и применением последовательно двух фильтра Гаусса с  $\sigma = 5$  и  $\sigma = 10$ . После чего полученный результат повторно зашумлен «шумом RGB».

На рис. 5. представлены графики функций распределения исходного изображения рис.4а для осей x и y (два верхних графика). Графики функций распределения скаженного изображения рис.4б изображения подаваемого на вход данного алгоритма (два графика посередине) и графики разности между гистограммой исходного и уменьшенного изображений (два нижних графика). Последние два графика получены путем простого вычитания значения каждого элемента в гистограмме исходного изображения от соответствующего значения в



гистограмме уменьшенного изображения.

Рис.5 Гистограммы, полученные по изображению на рис.4.

Алгоритм хорошо зарекомендовал себя при работе с цветными изображениями, в которых нет явного преобладания одного из цветов или оттенков. В случае более монотонных изображений качество опознавания может быть ниже. Рассмотрим изображение представленное на рис.5а

Это пример монотонного изображения с преобладанием серого (или серебристого) цветов. Для этого изображения алгоритм нашел «похожие» результаты из базы данных представленные рис.5б



Рис.5. Результат работа алгоритма на монотонных изображениях.

При этом стоит отметить, что неверный результат не обладал постоянством в том смысле, что при подаче на вход алгоритма исходного изображения (рис.5а) подвергнутого шумовым искажениям неверный результат (рис.5б) исчезал при одних параметрах шума, вновь возникая при других.

### **Выводы**

Предложен, реализован и протестирован алгоритм идентификации всех копий картинки по ее содержанию на основе использования гистограмм. Алгоритм показал хороший результат поиска цветных изображений идентичных входному, и содержащихся в специально разработанной базе данных изображений. Входные цветные изображения подвергались различного рода искажениям, таким как: изменение масштаба изображения, изменение ориентации изображения на 90 и 180 градусов, а также зашумление. Алгоритм продемонстрировал успешную работу в случае с входными изображениями, повернутыми на 90 или 180 градусов, а также уверенную работу при подаче на вход уменьшенного или/и зашумленного изображения в широком диапазоне искажений.

Точность предложенного подхода можно увеличить с использованием корреляционного сравнения гистограмм двух изображений. Однако данный подход будет влиять на продуктивность работы алгоритма с точки зрения времени выполнения, что в случае большой базы данных изображений может быть весьма существенным.

В работе с монотонными изображениями алгоритм продемонстрировал несколько худшую работу — помимо правильно найденного изображения результат работы периодически содержал изображения абсолютно не похожие на исходное.

### **Литература**

1. Яневичь Ю.М. «Задачи приема сигналов и определения их параметров на фоне шумов»: курс лекций / Ю.М. Яневичь - СПб.
2. Бабак В.П., Хандецкий В.С., Шлюфер Е. Обработка сигналов. - Киев, из-во «Лебедь», 1999. - 493с.
3. How does TinEye work? Режим доступа: <http://www.tineye.com/faq#how>