М. В. Стоцкий, А. А. Ардентов

Поиск похожих изображений для снимков DMSP

Научный руководитель: к.х.н. А. А. Московский

Аннотация. Данная работа посвящена разработке программного обеспечения для осуществления ранжирования изображений по наибольшей степени визуального сходства. Для сравнения двух изображений используются вектора–индексы, которые вычисляются для каждого из снимков с использованием вейвлет–преобразования. В статье описан параллельный алгоритм поиска K ближайших соседей для заданного вектора–индекса, который использует kd–деревья.

1. Введение

Научные работы, такие как метеорологические исследования, генерируют терабайтные базы данных [1]. Данные в таких базах обычно многомерные. Они должны быть визуализированы и исследованы для того, чтобы можно было найти интересующие объекты или извлечь значимые или качественно новые отношения. Один из самых ранних и наиболее продуманных примеров — SkyServer для Sloan Digital Sky Survey (SDSS) [2], который создает подробную цифровую карту большой части видимой вселенной и хранит несколько терабайт данных в общедоступном архиве. Многие статистические алгоритмы, требуемые для подобных задач, работают довольно быстро при обработке небольших по памяти множеств, но при работе с большими базами данных, не помещающихся в памяти, появляются заметные вычислительные трудности.

Данная статья посвящена исследованию одного из методов быстрой обработки и поиска визуальных данных.

Быстрая разработка технологий, в особенности на компьютерном железе и устройствах микроэлектроники, коренным образом изменяют большинство естественных наук с резким увеличением масштабов измерения. Мы могли бы выбрать наши примеры из почти любой дисциплины, но в данной работе будем работать с метеорологическими данными. Наша рабочая база данных — восьмидесятимерное пространство индексов, характеризующие контур и текстуру изображений. Изображения — снимки, полученные с космических спутников.

Каждый снимок представляется двумя изображениями — снимок в инфракрасном и видимом спектрах.

За время полета спутников накопилось огромное число изображений (снимков), которые необходимо обрабатывать при поиске похожих погодных условий для заданной местности. В связи с этим возникает необходимость разработать специальный алгоритм быстрого поиска.

2. Алгоритм индексирования

Для определения близости двух снимков необходимо ввести адекватную, с точки зрения метеоролога, метрику. Для этого каждому изображению ставится в соответствие некоторый вектор фиксированной длины. Эти вектора задают пространство индексов. После чего мы вводим некоторую функцию над множеством индексов, которая определяет метрику на заданном пространстве, например Евклидову [3].

Земная атмосфера имеет свойства высокотурбулентной системы, что позволяет естественным образом применить вейвлет–преобразования [4] для описания структур погодных данных, полученных по снимкам с метеорологических спутников. Полученные коэффициенты вейвлет–преобразования, формирующие вектора пространства индексов, описывают текстуру и форму структуры облаков для каждого изображения. Далее эти данные используются для интерактивного поиска изображений, основанного на схожести коэффициентов вейвлет–преобразования.

Вейвлет–преобразования широко используется в современных алгоритмах компрессии изображений; позволяет значительно (до двух раз) повысить степень сжатия чёрно–белых и цветных изображений при сравнимом визуальном качестве по отношению к алгоритмам предыдущего поколения, основанным на дискретном косинусом преобразовании, таких, например, как JPEG.

2.1. Сравнение форм

Для описания формы в данной работе мы применили метод центральных моментов. Для описания текстуры изображения используется метод обобщенных Гауссовых плотностей. В обоих случаях — и центральные моменты, и гауссовы плотности — вычисляются для

пирамидального разложения изображения в виде «пирамиды» подобных ему изображений с все уменьшающимся масштабом.

Отображение распределения яркости по масштабам на вектор характеристик формы изображения будет выполняться через центральные моменты амплитуд вейвлет–коэффициентов таким образом, что может быть измерена степень схожести формы. Как вариант, можно применить метод моментов к границам — протяженным линиям, на которых происходит резкое изменение яркости.

Для двумерной действительной функции $f(x, y)$ в конечном регионе S момент $(p + q)$–го порядка может быть представлен как

$$m_{p,q} = \iint_S x^p y^q f(x, y) dxdy.$$

Для дискретного изображения будем вычислять моменты как

$$m_{p,q} = \sum_{(i,j)\in S} i^p j^q f(i, j).$$

Используя теорию алгебраических инвариантов, возможно найти определенные функции моментов, которые остаются неизменными при преобразованиях изображений таких, как сдвиг, поворот и масштабирование. Например, для преобразования сдвига $x' = x + \chi$, $y' = y + \Psi$ центральные моменты

$$m_{p,q} = \iint_S (x - \bar{x})^p (y - \bar{y})^q f(x, y) dxdy$$

являются инвариантами; здесь $\bar{x} = \frac{m_{1,0}}{m_{0,0}}$, $\bar{y} = \frac{m_{0,1}}{m_{0,0}}$ обозначают координаты центра масс изображения. Для сдвига и зеркального отражения инвариантными функциями центральных моментов будут:

(1) для моментов первого порядка, $\mu_{1,0} = \mu_{0,1} = 0$ (всегда инвариантны);
(2) для моментов второго порядка, $(p + q) = 2$, инвариантами являются

$$\theta_1 = \mu_{2,0} + \mu_{0,2},$$
$$\theta_2 = (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2,$$
$$\dots$$

Итак, предлагаемый алгоритм выделения характеристик формы представлен следующим образом. Для каждого спутникового изображения выполняется:

(1) вейвлет-декомпозиция;

(2) сохранение в виде характеристик формы в базе данных, вычисленных нормализованных центральных моментов на всех масштабах.

Для простоты в качестве расстояния между центральными моментами векторов будем использовать Евклидово расстояние [3] или расстояние Махаланобиса [5].

2.2. Сравнение текстур

Для описания «текстуры» погодной турбулентности будем использовать статистические параметры распределения коэффициентов вейвлет–преобразования при различных масштабах. В случае, если эти распределения при различных масштабах можно считать «независимыми», их можно смоделировать обобщенными Гауссовыми плотностями

$$p(x, \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(\frac{1}{\beta})} e^{-(\frac{|x|}{\alpha})^{\beta}},$$

где $\Gamma(\cdot)$ — гамма–функция, α — величина, моделирующая ширину пика плотности распределения (среднеквадратичное отклонение), β обратно пропорционально скорости спада. Для моделирования межмасштабных зависимостей вейвлет–коэффициентов, вероятно, необходимо расширить пространство обобщенными Гауссовыми плотностями параметров и применить скрытую Марковскую модель [6].

В обоих случаях в качестве меры схожести между параметрами характеристик текстуры изображения θ_1 и θ_2 будет использоваться расстояние Кульбака–Лейблера [7], которое характеризует взаимную энтропию соответствующих плотностей $p(x, \theta_1)$ и $p(x, \theta_2)$:

$$D_{KL}(p(x, \theta_1)||p(x, \theta_2)) = \int p(x, \theta_1) \log \frac{p(x, \theta_1)}{p(x, \theta_2)} dx.$$

Используя логарифм по основанию 2, расстояние Кульбака–Лейблера дает в битах взаимную информацию между двумя изображениями.

Таким образом, при выделении характеристик текстуры, для всех изображений выполняется следующее:

(1) вейвлет–декомпозиция;
(2) вычисление обобщенных Гауссовых плотностей параметров и сохранение их как текстурных характеристик в базе данных.

2.3. Совместный анализ формы и текстуры

Совместный анализ форм и текстур является нетривиальной задачей, и во многом зависит как от характера изображений, так и от «типичных» задач, в которых он используется.

Можно отметить, что для двух различных типов изображений, а именно редкие облака и сильная облачность на фоне блика в телескопе, описанный подход работает и подбирает достаточно схожие снимки.

Последовательная версия программы, реализующей построение индексов, была написана в программной среде Matlab. Так как обработка изображений занимает достаточно много времени, на основе последовательной реализации была разработана параллельная версия программы, которая строит пространство индексов для набора космических снимков. Программа разбивает множество картинок на группы и каждая группа обрабатывается отдельно на разных вычислительных узлах кластера. Тем самым реализуется параллелизм по данным, при котором работа индексирующей программы ускоряется линейно относительно количества используемых узлов. Результатом выполнения алгоритма является множество векторов, каждый из которых соответствует некоторому снимку.

Построив пространство индексов, необходимо научиться на лету определять наиболее близкие вектора к заданному, тем самым определять наиболее похожие изображения для данного. Полный перебор при поиске близких изображений не допустим, т.к. мощность множества индексов очень велика.

3. Структуризация данных. Поиск

3.1. Деление на подмножества

Для того чтобы уменьшить количество операций во время работы алгоритма поиска, был выбран принцип, при котором происходит разбиение пространства индексов на подмножества. Алгоритм разбиения устроен так, что каждому вектору из пространства индексов ставится в соответствие целое число, тем самым два разных вектора, которым соответствует одинаковое число, принадлежат одному и тому же подмножеству.

Разбиение множества индексов осуществляется с использованием бинарных kd–деревьев [8]. Структура данных kd–дерево позволяет

быстро находить необходимое подмножество для данного изображения.

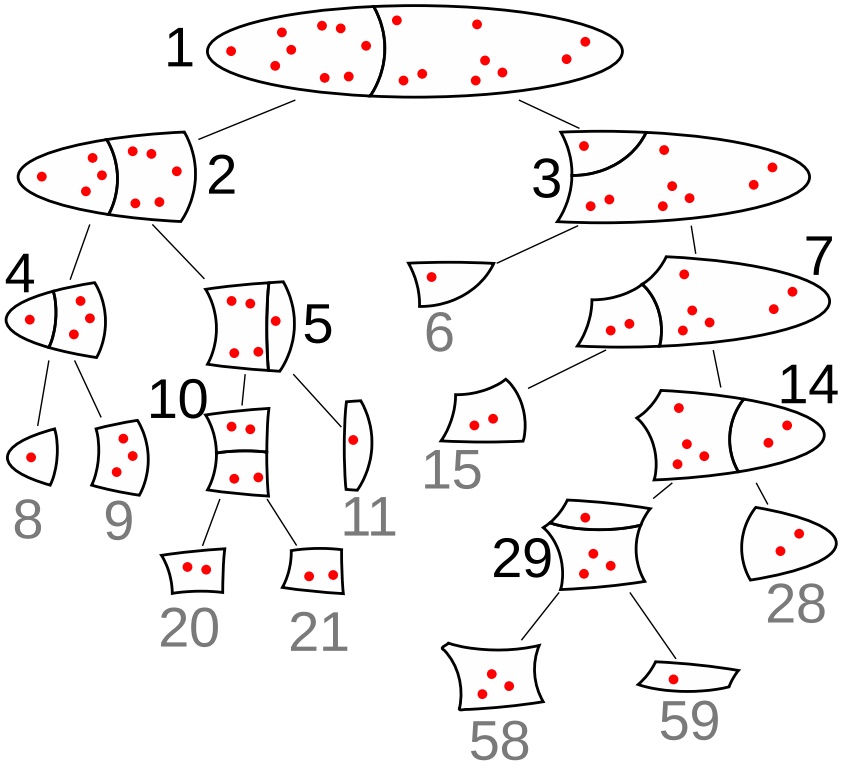

Рис. 1. пример kd–дерева

В каждой вершине kd–дерева имеется информация о разбиении множества векторов индексного пространства, которая представляет из себя номер узла (n), некоторое множество векторов и неравенство

$$x_i < C,$$

где i — это номер координаты, по которой производится разбиение; C — «место разреза». Вектора, которые удовлетворяют неравенству,

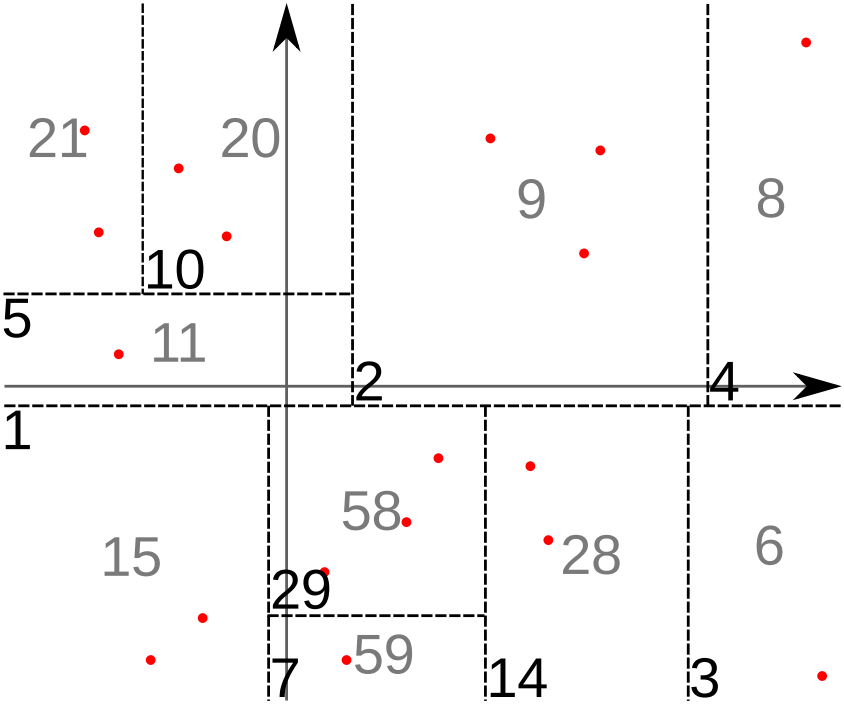

Рис. 2. пример разбиения

«спускаются» по левой ветке в узел с номером $2n$, а остальные — по правой с номером $2n + 1$. Корневой вершине (узел с номером 1) соответствуют все вектора индексного пространства. Тем самым, имея исходное множество векторов, всегда можно восстановить подмножество для любого из узлов kd–дерева.

При построении kd–дерева число i (номер координаты) выбирается для каждой из вершин дерева таким образом, чтобы разброс значений по этой координате был наибольшим. Разброс вычисляется как разность между максимальным и минимальным значениями координаты по всем векторам из подмножества. Далее все вектора из подмножества проецируются на i-ю ось пространства векторов, получаем прямую с точками, количество которых равно мощности подмножества. После этого выбираются две соседние точки a и b,

расстояние между которыми наибольшее, и вычисляется число C, которое равно $(a + b)/2$. Разбиение множества векторов продолжается до тех пор, пока мощность каждого из подмножеств не будет превышать некоторого заданного числа N. В листовых вершинах дерева окажутся искомые подмножества множества индексов.

На рис. 1 показан пример kd–дерева, которому соответствует разбиение множества векторов двумерного пространства индексов на подмножества, изображённое на рис. 2.

В результате получается, что алгоритм разбиения устроен таким образом, чтобы в первую очередь отсекать отдалённые вектора и группы векторов. Но при таком разбиении не гарантируется, что близкие вектора находятся в одном подмножестве. На основе данной структуры данных был разработан алгоритм поиска K ближайших соседей для некоторого вектора.

3.2. Поиск ближайших соседей

Для поиска K ближайших векторов для определённого вектора $v0$ используется список L, элементами которого являются пары

$$(n, v),$$

где n — номер вершины в kd–дереве; v — кратчайший вектор отложенный от конца вектора $v0$ до края области, ограничивающей подмножество, которому соответствует номер n. Список L всегда поддерживается в отсортированном по длине вектора v состоянии. Следовательно, первым элементом списка L всегда является ближайшее для исходной точки (вектора $v0$) подмножество векторов. Изначально список L состоит из одного элемента:

$$(1, (0, 0, ...)).$$

Алгоритм поиска заключается в следующих шагах (результат работы сохраняется в списке $neighs$, который изначально пуст):

(1) Если список L пуст, то окончить работу. Найдено K ближайших соседей.
(2) Иначе
   (a) Взять первую пару (n, v) из списка L.
   (b) Удалить первый элемент из списка L.
   (c) Если узел n является листовой вершиной в kd–дереве, то

(i) Если расстояние до самого дальнего соседа больше длинны v или длина списка $neighs$ меньше K, то
  (A) Отсортировать все элементы подмножества с номером n по расстоянию от $v0$ и записать их в список $neighs\_$.
  (B) Объединить списки $neighs\_$ и $neighs$ при помощи сортировки слиянием. Взять первые K элементов из объединения.
(ii) Иначе конец алгоритма. Найдено K ближайших соседей.

(d) Иначе
(i) Получить по номеру класса номер координаты, по которой был произведен разрез, и «место разреза».
(ii) Пересчитать вектора v_1 и v_2 для подмножеств $2n$ и $2n + 1$.
(iii) Вставить пары $(2n, v_1)$ и $(2n + 1, v_2)$ в список L, сохраняя список L отсортированным.

(3) Повторить процедуру.

Другими словами, мы перебираем точки из ближайших к $v0$ подмножеств до тех пор, пока объединение этих подмножеств не охватит шар с центром в точке $v0$, в котором окажется K точек.

Такой алгоритм поиска гарантирует, что будет найдено K ближайших векторов для заданного $v0$. В тоже время благодаря kd–дереву резко сокращается количество операций при поиске соседей по сравнению с полным перебором.

В тоже время была создана параллельная версия программы, реализующей поиск близких векторов. Самой тяжёлой операцией (требующей наибольшее количество действий) является сортировка элементов подмножества. Эта часть программы была оформлена в виде функции, которую можно вычислять на различных узлах кластера параллельно.

Также при поиске похожих изображений есть возможность учитывать только некоторые свойства картинок. Сейчас при поиске можно учитывать (или не учитывать) любые комбинации следующих свойств: координаты снимка, текстура или контур изображения, свойства только для видимого или только для инфракрасного изображения.

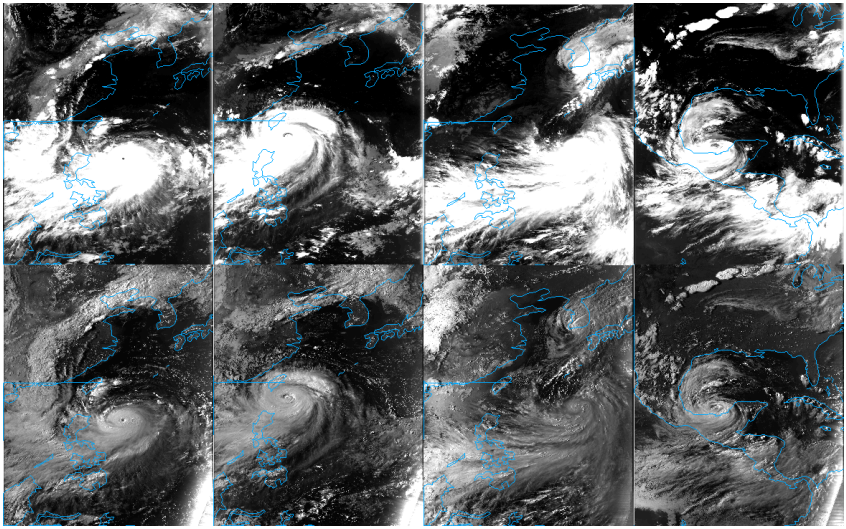

Рис. 3. Пример запроса

На рис. 3, 4 и 5 изображены результаты работы программы, реализующей алгоритм поиска похожих снимков. Самая левая пара изображений (снимки в инфракрасном и видимом спектрах) — это то изображение, для которого был заказан поиск похожих снимков. Вторая пара изображений — наиболее похожая пара на первую. То есть снимки отсортированы в порядке убывания визуального сходства слева направо. На примерах видно, что методы индексирования и поиска изображений дают хорошие результаты. Найденные изображения визуально очень похожи на искомые снимки. Также с помощью данного подхода успешно находятся испорченные снимки. Под испорченными снимками понимаются такие изображения, часть которых недоступна (часть снимка залита чёрным цветом). Испорченные снимки, возможно, могут возникать из-за обрывов связи со спутниками или различных помех, проявляющихся во время передачи снимков в базу данных на Земле.

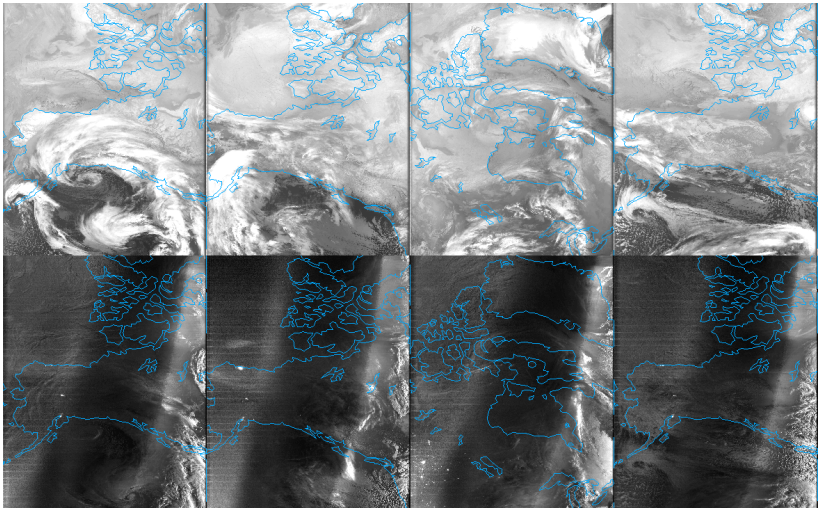

Рис. 4. Пример запроса

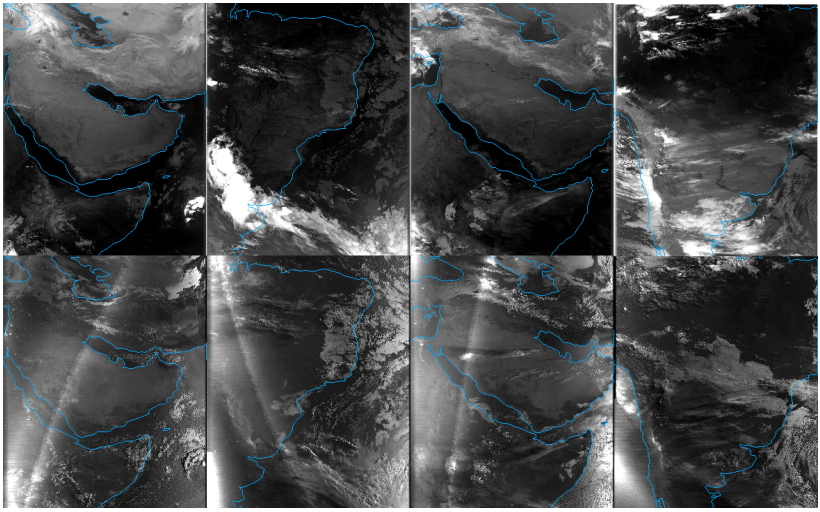

Рис. 5. Пример запроса

4. Заключение

В ходе исследования был разработан алгоритм поиска визуально схожих изображений с использованием вейвлет преобразования. Алгоритм реализован в системе просмотра архива космических снимков поверхности Земли для проекта Defense Meteorological Satellite Program (DMSP) [9].

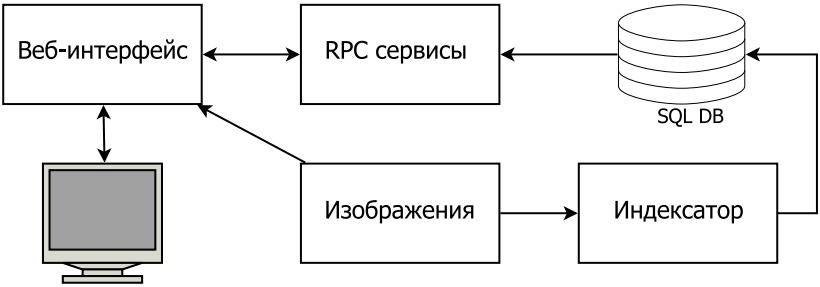

Рис. 6. Диаграмма веб–интерфейса

Для того, чтобы было удобно просматривать и анализировать работу алгоритма, был модифицирован веб-интерфейс. Дополнение заключается в добавлении функции показа схожих изображений. То есть была добавлена специальная возможность в браузер изображений, при использовании которой происходит поиск похожих на текущий снимков с последующим выводом результата на экран пользователю. Программа, реализующая алгоритм поиска похожих изображений, оформлена в виде web–сервиса в среде tomcat.

Список литературы

[1] I. Csabai M. T. (G. Herczegh) Spatial Indexing of Large Multidimensional Databases. ↑1

[2] SDSS SkyServer DR6. — http://cas.sdss.org/dr6/en/. ↑1

[3] Wikimedia Foundation; Wikipedia the free encyclopedia Euclidean distance. — http://en.wikipedia.org/wiki/Euclidean_distance. ↑2, 2.1

[4] Владимир Иванович Воробьев В. Г. Г. Теория и практика вейвлет–преобразования. ↑2

[5] Wikimedia Foundation; Wikipedia the free encyclopedia Mahalanobis distance. — http://en.wikipedia.org/wiki/Mahalanobis_distance. ↑2.1

[6] Владимир Николаевич Потапов Ю. Л. О. Марковские модели. ↑2.2
[7] Математическая статистика Боровков А.А.. — Москва: Наука, 1984. — 472 с. ↑2.2
[8] Препарата Ф. Ш. М. Вычислительная геометрия: введение. — Москва: Мир, 1989. ↑3.1
[9] NASA G. S. F. C. Defense Meteorological Satellites Program (DMSP) series. — http://heasarc.nasa.gov/docs/heasarc/missions/dmsp.html. ↑4

УГП, 5М41

M. V. Stotsky, A. A. Ardentov. *Searching similar images for DMSP pictures* // Proceedings of Junior research and development conference of Ailamazyan Pereslavl university. — Pereslavl, 2009. — p. 170–182. (*in Russian*).

Abstract. This paper treats a software development for ranking of images which based on the content-based image retrieval. We use vectors of indexes for the comparison of images (vector are calculated for each image with the help of Wavelet–transformation). Parallel algorithm of the search K nearest neighbors for the given vector uses kd–trees.