ПОИСК ПОХОЖИХ ИЗОБРАЖЕНИЙ

А.А. Ардентов, М.В. Стоцкий

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 найти интересующие объекты или чтобы извлечь значимые или качественно новые отношения. Один из самых ранних и наиболее продуманных примеров - SkyServer для Sloan Digital Sky Survey (SDSS) [2], которая 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 оного 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 - восьмидесяти мерное пространство индексов, характеризующие контур и текстуру изображений. Изображения - снимки, полученные с космических спутников.

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 является высоко турбулентной системой, 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 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:

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

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, можно 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

$$m_{p,q} = \iint_S (x - \overline{x})^p (y - \overline{y})^q f(x, y)\, dx\, dy$$

являются инвариантами; здесь $x = \frac{m_{1,0}}{m_{0,0}}, y = \frac{m_{0,1}}{m_{0,0}}$ 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

$$\theta_1 = \mu_{2,0} + \mu_{0,2}$$

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

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

1. вейвлет-декомпозиция;
2. вычисление нормализованных центральных моментов на всех масштабах и сохранение их в виде характеристик формы в базе данных.

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

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

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 Кулбака–Лейблера дает в битах взаимную информацию между двумя изображениями.

Таким образом, при выделении характеристик текстуры, для каждого изображения 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 «типичных» 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 алгоритм работает и подбирает достаточно схожие снимки.

Последовательная версия данного алгоритма реализована в программной среде Matlab. Но так как обсчет занимает достаточно много времени. Например, снимки, полученные со спутника в течении месяца, индексируются примерно за сутки (процессор 2.80GHz, 1000 Mb ОЗУ). Поэтому на основе последовательной реализации была разработана параллельная версия программы, которая строит пространство индексов для набора космических снимков.

Программа разбивает множество картинок на группы и для каждой группы запускает алгоритм индексирования на разных вычислительных узлах кластера. Тем самым реализуя параллелизм по данным, при котором данный алгоритм ускоряется линейно относительно количества используемых узлов. Результатом выполнения алгоритма является нормированное множество векторов.

Построив пространство индексов необходимо научиться на лету определять наиболее близкие вектор 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

Для того чтобы уменьшить время работы алгоритма поиска, был выбран принцип, при котором происходит разбиение пространства индексов на подмножества. Разбиение осуществляется таким образом, что близкие вектора попадают в одно подмножество. Следовательно, похожие между собой изображения попадают в один класс. 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. Описание алгоритма разбиения:

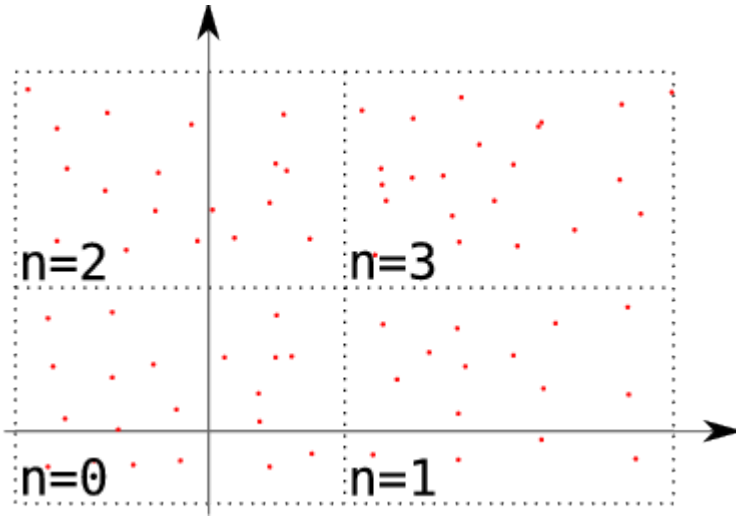

Рис. 1: пример разбиения

Поиск максимального и минимального значения по всем векторам пространства индексов для всех координат. Строится вектор максимальных (M) и минимальных значений (m).

- Вычисление вектора средних (h) значений по формуле h = $\frac{1}{2}$ (M +m).
- Вычисление номера (n) подмножества для каждого вектора.
  - Выбирается вектор (v) из пространства индексов.
  - n = 0
  - Если i-я координата вектора h больше или равна i-ой координаты вектора v, то в двоичной записи числа n в i-ой позиции ставится единица. Если i-я координата вектора h меньше i-ой координаты вектора v — ноль.

Иными словами

$$\begin{cases} n_i = \begin{cases} 1, h_i \geq v_i \\ 0, h_i < v_i \end{cases} \\ n = \sum_{i=0}^{|v|} 2^i n_i \end{cases}$$

На рисунке 1 показан пример разбиения множества векторов двумерного пространства индексов на подмножества.

Алгоритм поиска похожих изображений для данного сводится к определению номера подмножества, к которому принадлежит вектор, соответствующий данному изображению. Но при таком разбиении появляется серьезный недостаток — совсем не обязательно близкие между собой вектора должны принадлежать одному и тому же подмножеству. Следовательно, приведенного выше разбиения не достаточно для того, чтобы построить эффективный алгоритм поиска похожих

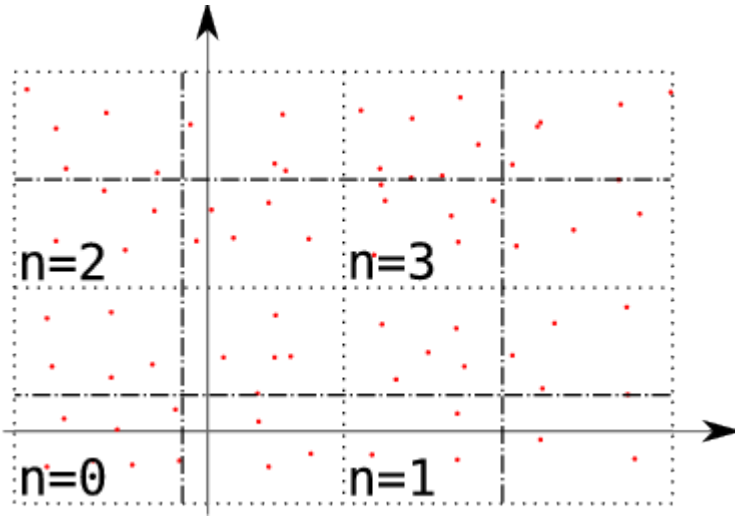

Рис. 2: пример расширенного разбиения

изображений. Попробуем дополнить текущие разбиение новым. Возьмем тоже самое разбиение, но сдвинем его по каждой размерности. Как показано на рисунке 3 теперь по каждой размерности вектора делятся не на две группы, а на три. Тем самым новое разбиение покрывает границы старого.

Теперь алгоритм поиска похожих изображений для данного заключается в определении номеров двух подмножеств, к которым принадлежит вектор для данного изображения. Если взять объединение двух этих множеств, то резко сокращается область поиска.

4 ЗАКЛЮЧЕНИЕ

В ходе исследования был разработан алгоритм поиска визуально схожих изображений с использованием вейвлет преобразования. Алгоритм реализован в системе просмотра архива космических снимков Земли для проекта Defense Meteorological Satellite Program (DMSP) [8].

Рис. 3: Диаграмма веб–интерфейса

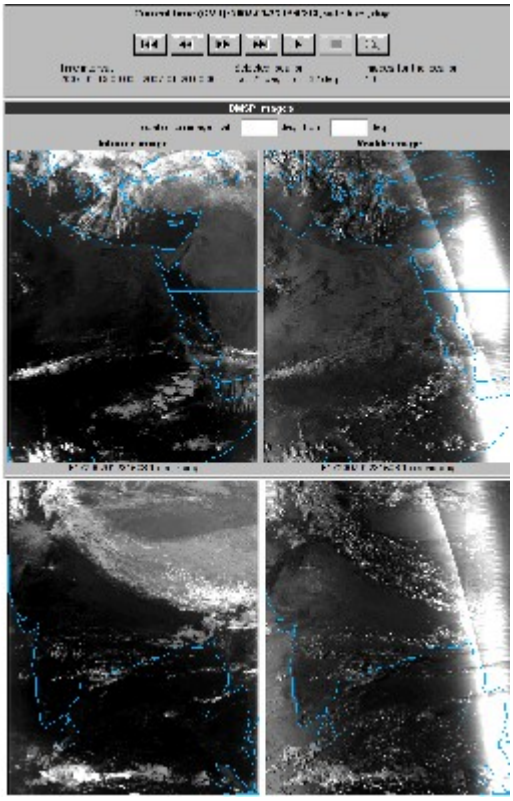

Рис. 4: Скриншот веб–интерфейса

Для того, чтобы было удобно просматривать и анализировать работу алгоритма, был модифицирован веб-интерфейс. Дополнение заключается в добавлении функции показа схожих изображений. Иными словами, была добавлена специальная кнопка в броузер изображений, при нажатии на которую происходит поиск похожих на текущий снимков с последующим выводом результата на экран пользователю.

БЛАГОДАРНОСТИ

М. Н. Жижин, А. А. Московский, Ф. А. Коряка

ЛИТЕРАТУРА

1. I. Csabai, M. Trencseni, G. Herczegh, L. Dobos, P. Jozsa, N. Purger, T. Budavari, A. Szalay, Spatial Indexing of Large Multidimensional Databases
2. SDSS, SkyServer DR6, http://cas.sdss.org/dr6/en/
3. Euclidean distance, http://en.wikipedia.org/wiki/Euclidean_distance
4. Владимир Иванович Воробьев, Вадим Геннадьевич Грибунин, Теория и практика вейвлет-преобразования
5. Mahalanobis distance, http://en.wikipedia.org/wiki/Mahalanobis_distance
6. Владимир Николаевич Потапов, Юрий Львович Орлов, Марковские модели.
7. Kullback–Leibler divergence, http://en.wikipedia.org/wiki/Kullback-Leibler_divergence

8. NASA, Defense Meteorological Satellites Program (DMSP) series, http://heasarc.nasa.gov/docs/heasarc/missions/dmsp.html