

Универсальная система визуализации геопривязанных данных из текстового файла на карте Яндекс

Н.Н. Воронина¹

ФГБУН ФИЦ «Морской гидрофизический институт РАН», Севастополь, Россия

¹ ORCID: 0000-0001-7301-2609, voronina.nataly@mail.ru

Аннотация

В 2016 году в №1 журнала «Научная визуализация» была опубликована статья [1], положившая начало дальнейшей серии публикаций, так или иначе связанных с визуализацией геопривязанных данных. Работа косвенно является продолжением этой серии, поскольку аналогично предыдущим статьям результат работы можно также увидеть на спутниковой карте Яндекс.

В статье достаточно подробно изложен пошаговый процесс разработки и отладки программного продукта, конечной целью которого является визуализация геофизических данных с их последующей выкладкой на спутниковую карту Яндекс в соответствии с теми географическими координатами, которые заданы в качестве входных. Программный продукт представляет собой полностью отлаженный законченный модуль, способный работать как в автономном режиме, так и в комплексе, направленном на выкладку полученного результата, на спутниковую карту Яндекс.

Ключевые слова: геопривязанные данные, визуализация, спутниковая карта Яндекс.

Введение

Существует множество примеров построения изображений на основе текстовых файлов, содержащих геопривязанные данные (данные, имеющие координаты привязки к карте). Для этих целей в различное время были созданы графические редакторы Surfer[2], MathCAD [3] и т.д., которые могут работать с текстовыми файлами на входе и обрабатывать эти файлы в том виде, который интересен пользователю. В данной статье мы не будем подробно останавливаться на каждом из них, у статьи другая цель, а именно – показать процесс создания изображения из любого текстового файла для последующей выкладки на спутниковую карту Яндекс без привлечения общеизвестных мощнейших графических редакторов.

АПИ (аббр. от англ. *Application Programming Interface* - описание способов взаимодействия одной компьютерной программы с другими) Яндекс [4] имеет весь необходимый функционал для реализации данного проекта, не смотря на некоторую ограниченность по сравнению с другими спутниковыми картами (Google Maps, например). Вдобавок Яндекс является отечественной разработкой, что имеет немаловажное значение. Авторы задались целью универсального формирования изображения на основе текстового файла, имеющего в своем арсенале только значения некоего параметра (будь то температура морской поверхности либо количество посетителей супермаркетов за период), а так же значения геопривязки (долгота/широта) параметра. То есть если речь идет о температуре морской поверхности, то в качестве геопривязки выступают широта и долгота той точки, в которой параметр температуры принимает именно это значение (полученное из наблюдений или прогностическое). Если же в качестве параметра выступает количество посетителей супермаркетов за период, то в таком случае в файле с геопривязанными данными будут использоваться долгота/широта точки с месторасположением интересующего супермаркета. Другими словами, долгота/широта

однозначно идентифицируют месторасположения значений того параметра, который будет показан на построенном изображении.

Необходимо отметить следующее допущение:

- в качестве входного может быть использован любой текстовый файл, не подвергавшийся предварительно никакой обработке графическими редакторами или чем-либо подобным, как например, файлы типа GRD (результат обработки редактором Surfer или Adobphotoshop [5]). В качестве программного средства для последующей реализации построения изображения был выбран скриптовый язык программирования PHP [6], так как он наилучшим образом встраивается в любой сайт, а именно сайт со спутниковой картой Яндекс в нашем примере, куда, в конечном счете, будет отправлено построенное изображение. Вдобавок PHP обладает всем спектром графических функций, необходимых для создания изображений.

Итак, конечную цель, а именно визуализацию произвольно заданных геопривязанных данных, будем реализовывать в несколько шагов:

1 построение черно-белого изображения,

2 «заливка» цветом построенного изображения,

3 выкладка изображения на спутниковую карту Яндекс.

Шаг 1 – построение черно-белого изображения

В качестве входных данных использован текстовый файл с расширением CSV, состоящий из трех столбцов:

- широта/долгота в первых двух столбцах;

- собственно значение параметра (в нашем примере это усеченная геострофика, то есть только одна из двух составляющих скоростей).

В данном случае расширение в названии текстового файла имеет значение только лишь для того, чтобы для открытия данного файла в программе PHP была использована функция, предназначенная именно для этого расширения.

Первоначально построим черно-белое изображение данных, содержащихся во входном файле. Черно-белое изображение строится только лишь на основе долготы/широты тех точек, в которых присутствуют значения параметра. Собственно, само значение параметра появится на этапе разработки в шаге 2, так как значения параметра на рисунках – это фактически цветовая палитра, в чем далее нам предстоит убедиться.

Результат работы скрипта на шаге 1 показан на рис.1. Размер холста вычисляемый и зависит от того, какое максимальное количество точек во входном файле представлено по широте и долготе:



Рис.1 Первичное черно-белое изображение входных данных

$x = (\text{max_do} - \text{min_do}) * \text{const}$,

$y = (\text{max_sh} - \text{min_sh}) * \text{const}$,

где:

max_sh , min_sh – значения максимальной и минимальной широт во входном файле,

max_do , min_do – значения максимальной и минимальной долгот во входном файле соответственно,

const – произвольная константа, задающая увеличение/уменьшение изображения для улучшения просмотра (своего рода масштаб).

Для того, чтобы получить это же изображение в сплошном черно-белом представлении, можно воспользоваться константой const , уменьшив наш результат, и увидеть на экране изображение, см. рис.2.



Рис.2 Первичное изображение входных данных в сплошном черно-белом представлении

Уточним – на рис.1 отчетливо видны точки, но тем ни менее для их построения была использована функция для построения эллипса `imagefilledellipse()` с заданием черного (временно) цвета в качестве заливки. Точка в данном случае выступает как частность эллипса, то есть точку при равнозаданных размерах радиусов считаем эллипсом (более универсальным функционально).

В скрипте PHP помимо вышеуказанной также были использованы на шаге 1 следующие функции:

`imagecreatetruecolor()` – создание изображения рассчитанного размера,

`imagecolorallocate()` – создание цветов (белого для построения рисунков на холсте и черного для заливки точек-эллипсов),

`imageFilledRectangle()` – построение на холсте прямоугольника белого цвета в качестве фона изображения.

На этом шаг 1 считаем полностью реализованным.

Шаг 2 - «Заливка» цветом построенного изображения

На данном этапе будем дополнительно использовать текстовые данные того же файла, но уже с файлом произвольной палитры типа RGB, то есть в файле-палитре присутствуют три столбца, соответствующие значениям Red, Green, Blue. Пока для наглядности палитру будем строить прямо на том же холсте, созданном ранее на шаге 1. Для этого увеличим размер холста, рассчитанного в шаге 1, на желаемую ширину палитры:

$- x = ((\text{max_do} - \text{min_do}) + \text{const_add_palett}) * \text{const}$,

где const_add_palett – желаемая ширина изображения палитры (например, от 0 до 10). Ноль в данном случае говорит об отсутствии палитры на холсте.

В будущем изображение палитры сделаем отдельно от основного изображения, чтобы иметь возможность размещать изображение палитры в любом месте за картой

вертикально или горизонтально по желанию исследователя, но это уже тема другой статьи.

Итак, необходимо каждую точку исходного, пока черно-белого изображения закрасить тем цветом из палитры, который соответствует значению параметра, полученного в точке с заданными долготой/широтой. Чтобы установить это соответствие, поступим следующим образом:

- подсчитаем общее количество точек со значениями параметра (в используемом нами примере 2993) и общее количество строк со значениями цветов в палитре, например 150;

- поскольку точек со значениями параметра существенно больше, чем цветов в палитре, 2993 против 150, а при выборе в будущем других входных данных это превышение в той или иной степени будет неизменно сохраняться, определим путем несложных вычислений какому цвету будет принадлежать диапазон значений параметра из входных данных;

- поменяем в указанной в шаге 1 функции построения эллипса `imagefilledellipse()` цвет заливки с черного на тот цвет параметра, соответствие которому мы уже определили выше.

На рис.3 показан результат работы обновленного скрипта PHP с отображением используемой палитры различной ширины (слева и справа), а также с указанием максимального и минимального значений параметра, соответствующих начальному и конечному цветам палитры.

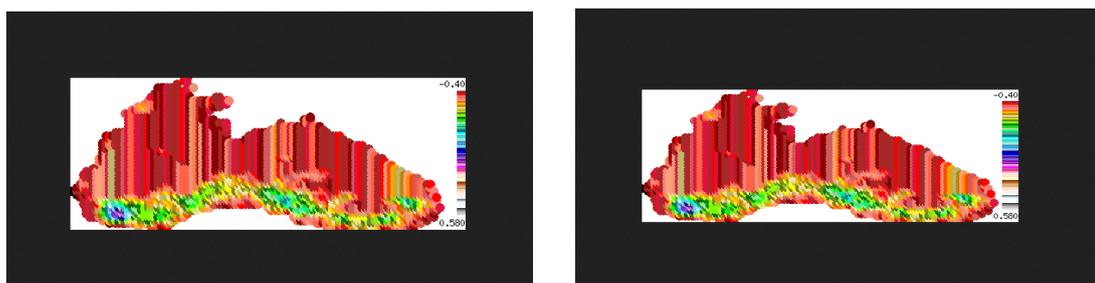


Рис. 3 Цветное изображение с палитрой различной ширины и надписями

Надо заметить, что указания нескольких значений параметра, помимо максимального и минимального, могут быть заданы по-разному – либо соответствующими надписями-делениями у самой шкалы палитры, либо в качестве «всплывающей» подсказки при «снятии» этих значений с холста аналогично «снятию» значений координат на спутниковой карте Яндекс [7], о чем должны быть сделаны некие допущения, но это уже предмет будущей работы, по завершении которой поговорим в другой статье.

В скрипте PHP помимо уже использованных ранее в шаге 1 были использованы в шаге 2 следующие функции:

`imagecolorallocate()` – создание различных цветов для построения палитры,

`imageFilledRectangle()` – построение на холсте прямоугольника со всеми цветами в качестве палитры,

`ImageString ()` – для создания надписей на изображении палитры.

Первые два шага будем считать предварительными перед выкладкой сформированного изображения на спутниковую карту Яндекс [8].

Шаг 3 - выкладка изображения на спутниковую карту Яндекс

В готовом изображении перед выкладкой на карту необходимо сделать еще одно – прозрачный фон. Но поскольку это требует добавления всего лишь одной функции `Imagecolortransparent()` для цвета фона (в нашем примере это белый цвет), а результат работы данной функции проявляется наилучшим образом именно после выкладки на

карту, то было решено добавить изображению прозрачность именно на шаге 3 – при выкладке на спутниковую карту.

Готовый скрипт PHP для построения изображения вставим в обрамление HTML. Передадим уже готовое изображение с необходимыми параметрами (координаты точек с параметром, центр будущей карты и т.д.) в скрипт JS [9], поскольку именно JS отвечает за построение спутниковой карты Яндекс. В результате получим изображение, представленное на рис. 4.



Рис.4 Изображение на спутниковой карте Яндекс

Теперь, когда все готово, в качестве эксперимента попробуем заменить файл с палитрой, используя для этого палитру с небольшим количеством цветов. Тогда то же изображение [10] будет выглядеть, как показано на рис.5



Рис.5 Изображение на спутниковой карте Яндекс с другой палитрой

Другими словами, в дальнейшем пользователь сам будет решать и соответственно выбирать, какую палитру ему удобнее использовать, но это станет предметом исследований в будущих публикациях.

И наконец, на рис.6 показан пример изображения осредненной за декаду температуры (в качестве параметра) поверхности Мирового океана на сайте Гидрометеоцентра

России [11], сделанный совершенно иными средствами. Данный пример можно сделать и нашим достаточно простым, изложенным в статье способом.

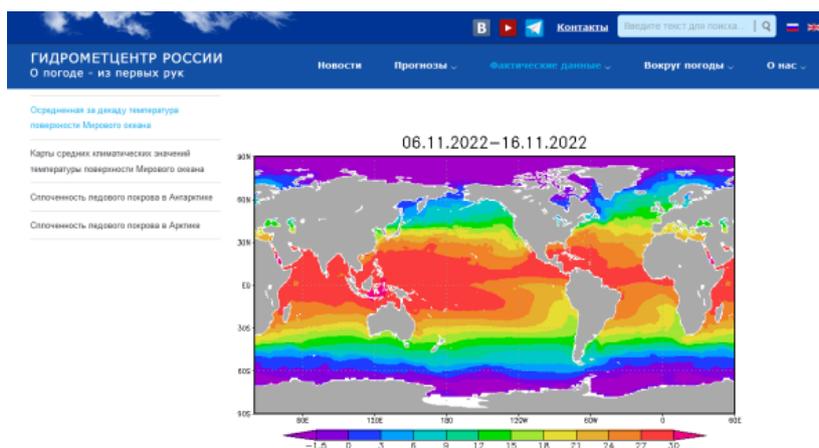


Рис.6 Фактические данные Гидрометеоцентра России

Заключение

Отметим, что данные работы с пошаговым построением изображения проведены несколько нетрадиционным методом, то есть отладка программы проводилась не отдельными модулями, а каждый новый шаг включал в себя отлаженный ранее, то есть происходило определенного рода «нанизывание». Таким образом, именно этот подход позволил прийти к конечной цели с уже целиком отлаженным, работающим скриптом PHP.

Данный стержневой, работающий скрипт будет служить главенствующей основой при наших дальнейших работах в направлении универсальности. Отметим, что эти дальнейшие работы уже ведутся. Универсальность будет заключаться в возможности выбора любого текстового файла, а так же файла с палитрой. И самое главное – предстоит сделать программный продукт по построению параметра скорости, то есть на изображение будут нанесены стрелки с указанием направления движения скоростных потоков.

Если обратиться к примерам общемирового уровня, то на сайте COPERNICUS [12], где представлено огромное количество изображений геофизических параметров, отметим, что программный продукт может найти применение и в системах-аналогах Copernicus. Также данный программный продукт может стать основой для работ в любом регионе на картографической поверхности. На сайте с картой Яндекс в дальнейшем предполагается создать различного рода выборки интересующих входных данных и палитр, что ляжет в основу темы для другой статьи.

Благодарности

Работа выполнена в рамках государственного задания по теме FNNN-2021-0004 «Фундаментальные исследования океанологических процессов, определяющих состояние и эволюцию морской среды под влиянием естественных и антропогенных факторов, на основе методов наблюдения и моделирования» (шифр «Океанологические процессы»)

Список литературы

1. Voronina N. N., Inyushina N. V., Mamchur N. L., Kryl' M. V. *Analiz i sopostavlenie programmnyh sredstv vizualizacii morskikh prognozov na osnove raschetnyh dannyh operativnoj okeanografii po chernomorskому bassejnu na primerah sevastopol'skogo i krymskogo regionov.*// Nauchnaya vizualizaciya. Elektronnyj zhurnal, ISSN 2079-3537, №

- 1, Т. 8, 2016. – s. 146 - 155. [Electronic resource]. URL: <http://sv-journal.org/2016-1/09.php?lang=ru>
2. Dogan T., Kastner P., Mermelstein R. *Surfer: a fast simulation algorithm to predict surface temperatures and mean radiant temperatures in large urban models* // Building and Environment. 2021. Т. 196. С. 107762.
3. Никулин Е. *Компьютерная 2d-графика. Программирование в MathCAD*// Лань, 2022
4. Коптенок Е. В., Савенко А. В., Фомин И. И., Трунников М. В., Сухарев Е. А. *Анализ возможностей сервисов для реализации виртуальных туров на примере использования API Яндекс.Карт.*// Молодой ученый. № 19 (309), 2020. - С. 130-133. URL: <https://moluch.ru/archive/309/69891/>
5. Фолкнер Э., Чавез К. *Adobe Photoshop CC. Официальный учебный курс* // Эксмо-Пресс, 2021.
6. Зандстра М. *PHP 8. Объекты, шаблоны и методики программирования* // Диалектика, 2021. [Electronic resource]. URL: <https://php.ru/manual/>
7. [Electronic resource]. URL: <http://innovation.org.ru>
8. Мохов В.А., Кубил В.Н., Кузнецова А.В., Георгица И.В. *Рекурсивный алгоритм синхронизации Api-запросов к гис-сервису Яндекс.Карты*//Фундаментальные исследования. № 9-1, 2015. – С. 33-38. URL: <https://fundamental-research.ru/ru/article/view?id=38961>
9. Foerderer J., Bender M., Heinzl A. *Regulation of digital platform ecosystems: evidence from russia's google vs yandex ruling.*// В сборнике: International Conference on Information Systems 2018, ICIS 2018. 39. 2018.
10. [Electronic resource]. URL: http://innovation.org.ru/generate_image_do-sh.php
11. Сервис Программы «Гидрометцентр России» // [Electronic resource]. URL: <https://meteoinfo.ru/ocean>
12. Сервис Программы «Коперникус» // [Electronic resource]. URL: <https://data.marine.copernicus.eu>

Universal System for Visualizing Geo-Referenced Data From a Text File on a Yandex Map

N.N. Voronina¹

FSBSI FRC Marine Hydrophysical Institute RAS, Sevastopol, Russia

¹ ORCID: 0000-0001-7301-2609, voronina.nataly@mail.ru

Abstract

In 2016, an article [1] was published in the journal Scientific Visualization No. 1, which marked the beginning of a further series of publications, one way or another related to the visualization of geo-referenced data. The work is indirectly a continuation of this series, since, similarly to previous articles, the result of the work can also be seen on the Yandex satellite map. The article describes in sufficient detail the step-by-step process of developing and debugging a software product, the ultimate goal of which is the visualization of geophysical data with their subsequent display on the Yandex satellite map in accordance with the geographic coordinates that are set as input. The software product is a fully debugged complete module that can work both offline and, in a complex, aimed at laying out the result on the Yandex satellite map.

Keywords: georeferenced data, visualization, Yandex satellite map.

References

1. Voronina N. N., Inyushina N. V., Mamchur N. L., Kryl' M. V. *Analiz i sopostavlenie programmnyh sredstv vizualizatsii morskikh prognozov na osnove raschetnykh dannykh operativnoy okeanografii po chernomorskemu bassejnu na primerah sevastopol'skogo i krymskogo regionov.* // Nauchnaya vizualizatsiya. Elektronnyy zhurnal, ISSN 2079-3537, № 1, T. 8, 2016. – s. 146 - 155. [Electronic resource]. URL: <http://sv-journal.org/2016-1/09.php?lang=ru>
2. Dogan T., Kastner P., Mermelstein R. *Surfer: a fast simulation algorithm to predict surface temperatures and mean radiant temperatures in large urban models* // Building and Environment. 2021. T. 196. C. 107762.
3. Nikulin Ye. *Kompyuternaya 2d-grafika. Programmirovaniye v MathCAD* // Lan, 2022
4. Koptenok Ye. V., Savenko A. V., Fomin I. I., Trunnikov M. V., Sukharev Ye. A. *Analiz vozmozhnostey servisov dlya realizatsii virtualnykh turov na primere ispolzovaniya API Yandeks.Kart.* // Molodoy uchenyy. № 19 (309), 2020. - S.130-133. URL: <https://moluch.ru/archive/309/69891/>
5. Folkner E., Chavez K. *Adobe Photoshop SS. Ofitsialnyy uchebnyy kurs* // Eksmo-Press, 2021.
6. Zandstra M. *PHP 8. Obekty, shablony i metodiki programmirovaniya* // Dialektika, 2021. [Electronic resource]. URL: <https://php.ru/manual/>
7. [Electronic resource]. URL: <http://innovation.org.ru>
8. Mokhov V.A., Kubil V.N., Kuznetsova A.V., Georgitsa I.V. *Rekursivnyy algoritm sinkhronizatsii Api-zaprosov k gis-servisu Yandeks.Karty* // Fundamentalnye issledovaniya. № 9-1, 2015. – S. 33-38. URL: <https://fundamental-research.ru/ru/article/view?id=38961>
9. Foerderer J., Bender M., Heinzl A. *Regulation of digital platform ecosystems: evidence from russia's google vs yandex ruling.* // V sbornike: International Conference on Information Systems 2018, ICIS 2018. 39. 2018.
10. [Electronic resource]. URL: http://innovation.org.ru/generate_image_do-sh.php

11. Servis Programmy «Gidromettsentr Rossii» // [Electronic resource]. URL:
<https://meteoinfo.ru/ocean>
12. Servis Programmy «Kopernikus» // [Electronic resource]. URL:
<https://data.marine.copernicus.eu>