

# Мультивременные слои в GeoMixer WEB-GIS

[Обсудить в форуме](#) Комментариев — 14

Эта страница опубликована в основном списке статей сайта по адресу <http://gis-lab.info/qa/geomixer-multitemporal-layers.html>

Общий обзор и детали реализации механизма работы с временными архивами данных в GeoMixer Web-GIS

## Содержание

- [1 Философское введение](#)
- [2 Как использовать в ГеоМиксере](#)
- [3 Примеры](#)
- [4 Как оно работает \(скучный технический раздел :\)](#)
- [5 Банальное заключение](#)
- [6 \(Бес\)полезные ссылки](#)

## Философское введение

Иногда данных бывает много, очень много. Как правило, пытаться отобразить их все — плохая идея. И дело тут даже не в технических сложностях и ограничениях, а в возможностях человека. Сложно представить себе ситуацию, когда пользователю одновременно нужно увидеть несколько десятков тысяч отдельных объектов.

Для работы с большими объёмами данных используют несколько приёмов: аналитическую предобработку данных (карта плотности вместо отдельных объектов), кластеризацию (вместо отдельных данных показываем суммированную информацию по ним), фильтрацию данных (показываем только интересные в данный момент объекты).

В нашей практике реализации проектов на основе платформы [GeoMixer](#) часто встречался особенный вид данных — объекты, накапливающиеся с течением времени. Это могут быть данные о погоде (температура, давление, сила и направление ветра), пожары, результаты мониторинга (рубки, нефтяные слики, и т.д.) и множество других примеров. Так же временной аспект могут иметь и сами снимки ДЗЗ (например, при регулярной съёмке одной и той же территории).

Изначально мы использовали стандартные механизмы работы с наборами данных: разбивали данные на отдельные слои по дате; объединяли всё в один слой, а потом фильтровали объекты слоя по временному атрибуту; делали специализированные интерфейсы под конкретные проекты. Однако все эти варианты имеют ряд технических сложностей, трудны в поддержке и не всегда адекватно выглядят для пользователей.

Со временем пришло понимание, что для подобных данных нужно отдельное решение — мультивременные слои. Мультивременный слой в ГеоМиксере — это векторный слой, объекты которого могут быть отображены за какой-то определённый промежуток времени. При этом не просто происходит фильтрация данных в момент визуализации, но и оптимизация передачи данных с сервера. Таким образом, становится возможным создавать слои практически с неограниченным количеством данных при условии, что при просмотре данных будет выбрана разумная пространственно-временная область.

## Как использовать в ГеоМиксере

Мультивременный слой в ГеоМиксере является частным случаем векторного слоя (из меню: Слой -> Создать векторный слой). Можно использовать готовый источник данных (например, shp файл) или создать пустой слой с заданием атрибутивных полей вручную. При этом среди атрибутов слоя должны быть поля временного типа.

После сохранения карты и перезагрузки страницы слева над деревом слоёв появится календарик для выбора

периода отображения. Его можно использовать для просмотра данных за определённый период. Стоит иметь в виду, что календарик по умолчанию действует на все слои в карте.

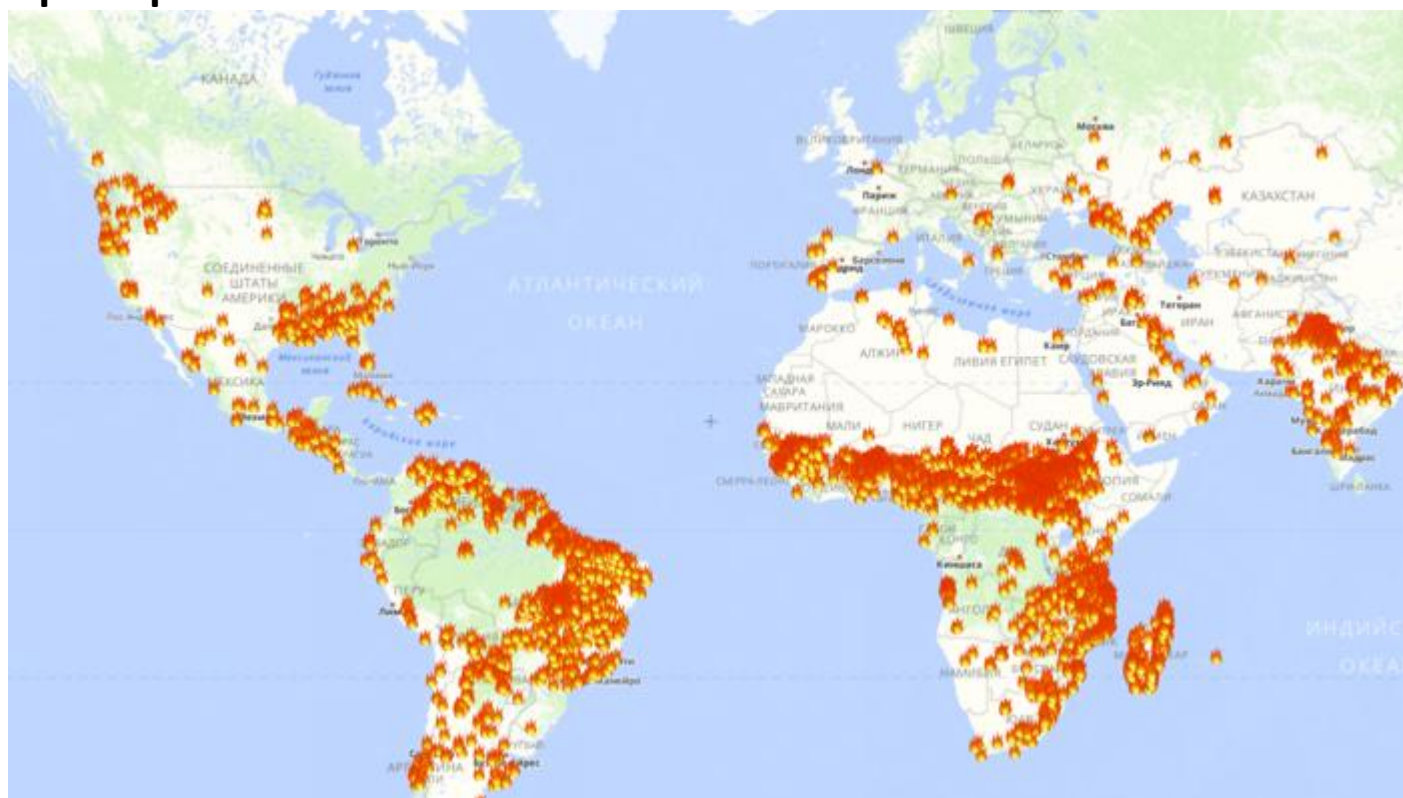
Для работы с мультिवременными слоями через GeoMixer API в класс [Layer](#) добавлен метод `setDateInterval()`:

```
function setDateInterval(dateBegin, dateEnd)
    dateBegin: Date - начальная дата периода отображения
    dateEnd: Date - конечная дата периода отображения
```

Функция позволяет отображать данные векторного слоя за период с `dateBegin` до `dateEnd`.

При этом через API для разных слоёв можно задавать разные периоды отображения.

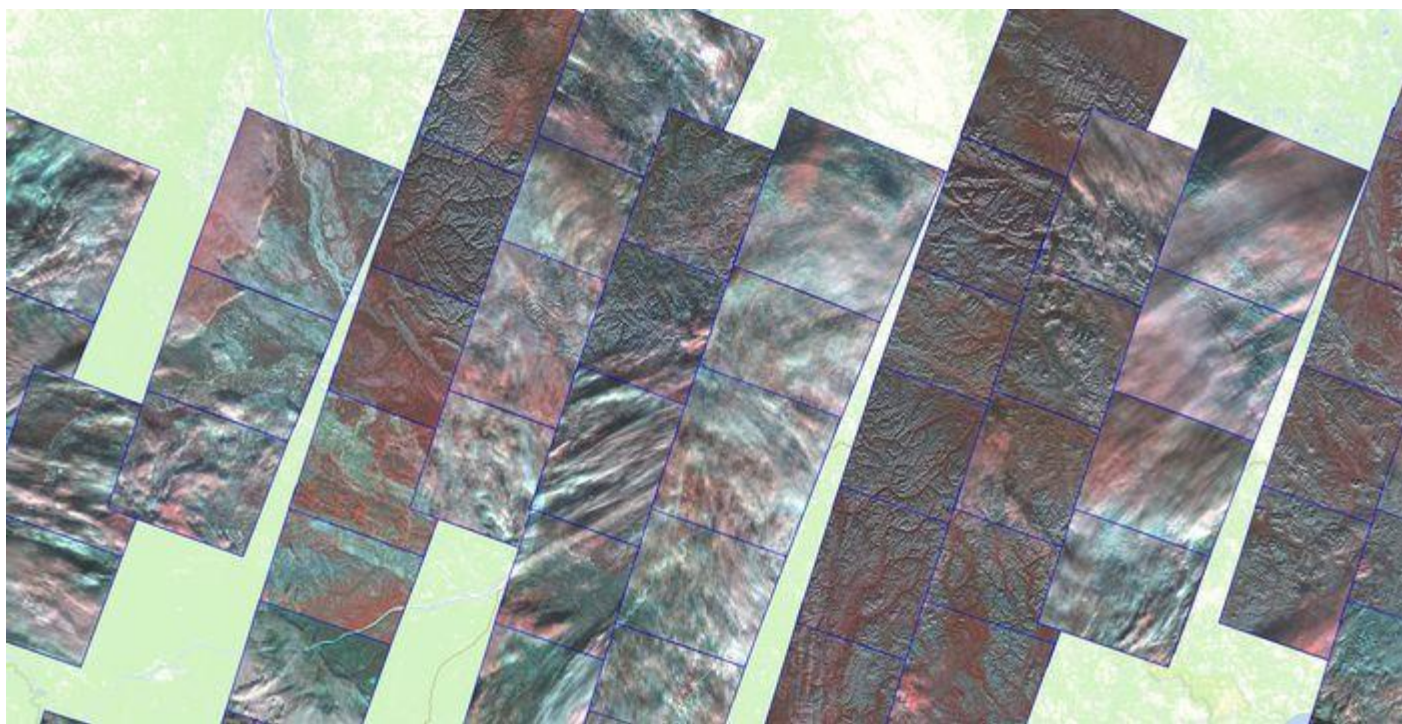
## Примеры



Наш [любимый векторный слой](#) — точки с температурными аномалиями (пожарами), полученными со спутников [TERRA](#) и [AQUA](#).

В слое 6.6 млн. точек с августа 2012 по текущий момент. На календарике сверху слева можно выбрать любой день и посмотреть пожары по всему миру за эту дату.

Более того, реализация мультिवременных слоёв в ГеоМиксере позволяет не только просматривать данные за один день, но и увидеть данные на локальную территорию за большой промежуток времени (лишь бы общий объём данных в этой пространственно-временной области был разумным). Вот пример такой выборки — [пожары за весь прошлый год на Московскую область](#).



Ещё пример — данные [Landsat 8](#). Данные открытые, мы скачиваем безоблачные сцены на территорию России и храним их в ГеоМиксере в виде мультитременного слоя. Например, вот [съёмка за одну неделю в ноябре](#) (каждый прямоугольник — это одна снятая спутником сцена). Кстати, если приблизиться, будут видны и сами снимки в полном разрешении (30м/пиксель)!

При помощи такого мультитременного слоя можно искать съёмку на конкретный район — нужно зазумиться в нужное место, выбрать период побольше и переключать слои кликами по ним. Вот [доступная съёмка Landsat 8 на Северобайкальск с апреля этого года](#) (кликайте для переключения снимков).

Так же мы экспериментируем с различными дополнительными элементами управления для анализа временной компоненты данных. [Вот пример](#) работы с одним из таких инструментов - "таймлайном". Этот инструмент упрощает понимание того, как данные из некоторой пространственной области распределены во времени.



Следующий пример — мониторинг объектов. У нас ведётся несколько мониторинговых проектов, данные по которым хранятся именно в форме мультитременных рядов.

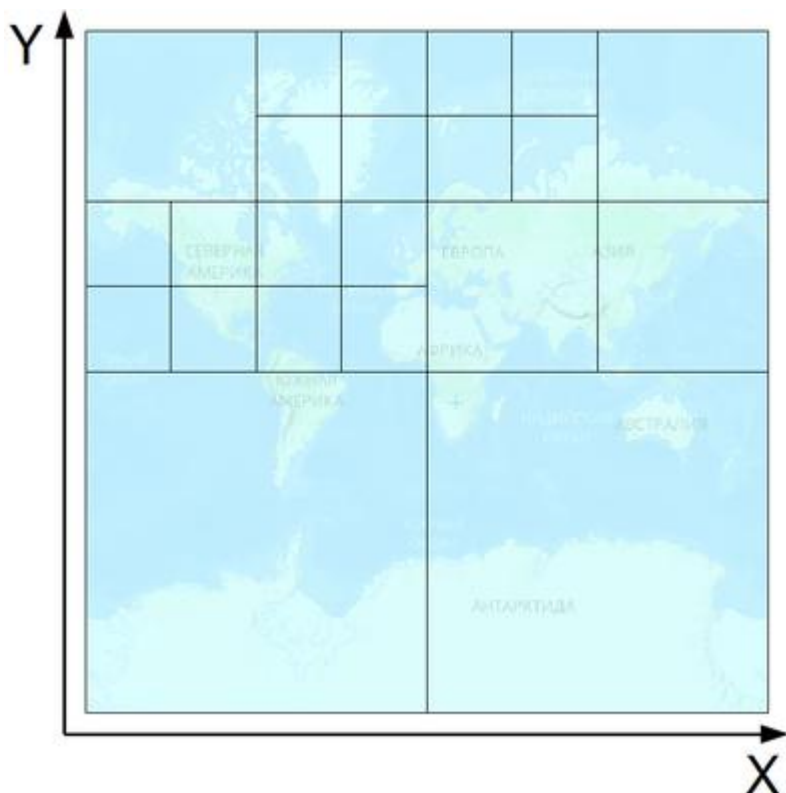
На скриншоте выше показан пример визуализации данных мониторинга судов за один день в Баренцевом



море (к сожалению, лицензия на данные не позволяет нам использовать их в публичных проектах). Каждый треугольник — зафиксированное положение корабля. Направление иконки соответствует направлению движения, а цвет — типу судна. В этом слое на данный момент 3.7 млн. объектов (положений кораблей). Использование мультिवременного слоя позволяет быстро просматривать данные за произвольный временной период.

## Как оно работает (скучный технический раздел :)

Основная проблема, которую нам пришлось решить — передача на клиент только нужных данных без существенной загрузки сервера. Можно было бы настроить в БД нужные индексы, при каждом запросе делать выборки, результаты конвертировать в понятный клиенту формат и отгружать в браузер. Но нам это показалось недостаточно эффективным решением. Кроме того, у нас есть хорошо себя показавшие векторные тайлы, которые хотелось применить и для мультिवременных слоёв. Но всё оказалось не так просто...

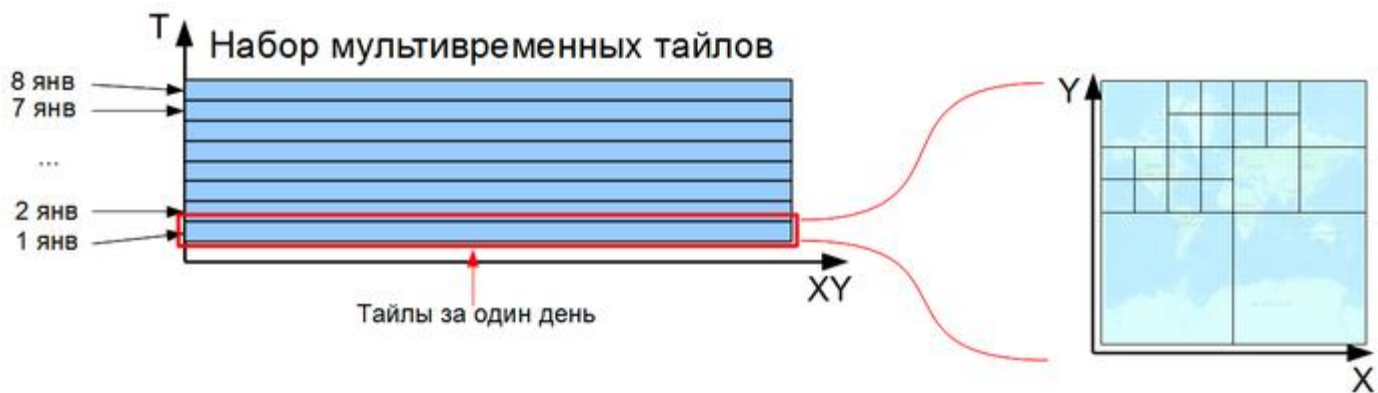


*Пример разбиения пространства на тайлы разного размера. Размер тайла определяется количеством попавших в него данных.*

Сначала пару слов про наши векторные тайлы. Для передачи на клиент векторных данных мы разбиваем их на квадратные кусочки (тайлы), сразу же преобразуем эти кусочки в GeoJSON-подобный формат и сохраняем на сервере. Клиенту просто отдаём статические файлы. Степень разбиения определяется размером данных — если в тайле оказывается слишком много объектов, он разбивается на 4 подтайла. Такой подход сохраняет баланс между количеством запросов на сервер и загрузкой лишних (не видимых в данный момент) данных.

Алгоритм построения тайлов для набора векторных объектов выглядит примерно так. Сначала вычисляется общий размер данных для всего набора. Если он меньше определённого порога, то всё записывается в один тайл, и процесс завершается. Если размер данных слишком большой, данные делятся на 4 части в соответствии со своим положением в пространстве. Для каждой части процесс повторяется.

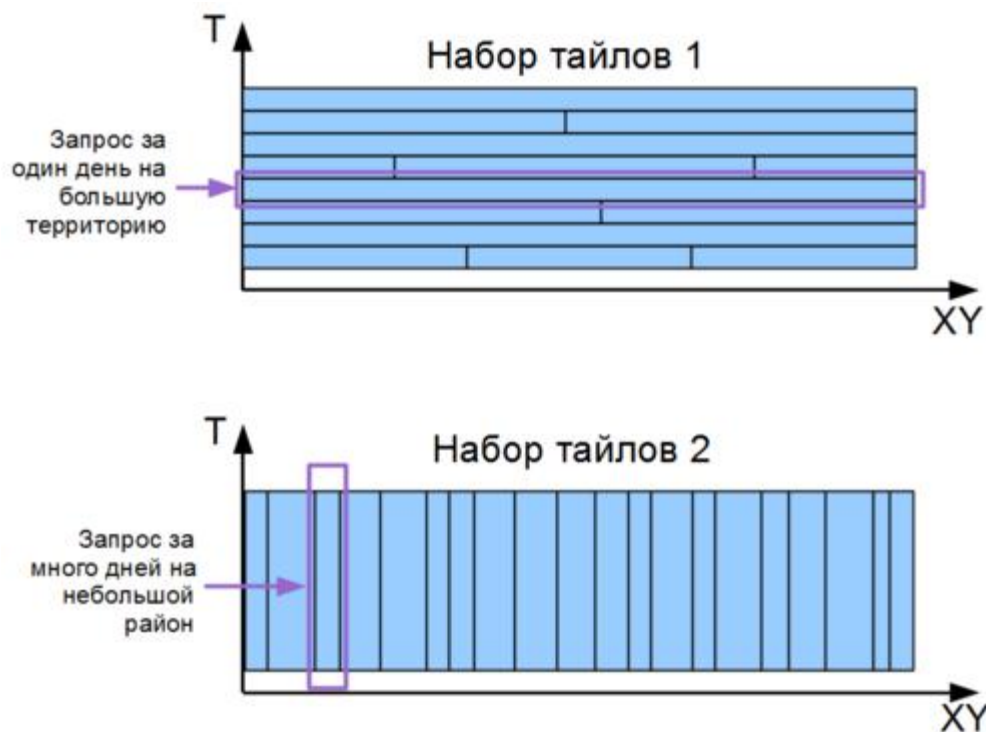
Однако при применении этой схемы к мультिवременным слоям мы столкнулись с проблемой формы пространственно-временной области, которая может быть запрошена.



*Мультивременные тайлы по дням. Дополнительно показана структура тайлов за один конкретный день. Каждый прямоугольник на графике слева — тайлы для данных за один конкретный день.*

Всё дело в том, что квадратные тайлы (векторные или растровые — не важно) хорошо подходят для передачи на клиент только при условии, что область просмотра (экран монитора) тоже примерно квадратная. Но представьте на минутку, что пользователь захочет отрисовать картинку размером  $1 \times 100000$  или  $100000 \times 1$  пикселей! Стандартные квадратные тайлы  $256 \times 256$  тут не подойдут — вам придётся передать ровно в 256 раз больше данных, чем нужно клиенту!

Именно такая ситуация возникает в случае с пространственно-временными данными — пользователи хотят смотреть и данные на весь мир за один день (аналог запроса  $10000 \times 1$ ) и данные за 10 лет на деревню бабушки ( $1 \times 10000$ ). Если есть тайлы только за отдельные дни, во втором случае придётся сделать минимум 3650 запросов; если в каждом тайле есть данные сразу за 10 лет, при просмотре всего мира за любой день мы загрузим весь архив (точнее говоря, попробуем загрузить...).



*Наборы тайлов с разным разрешением по времени. Показаны характерные запросы от пользователей, которые обрабатываются этими наборами. Каждый тайл содержит примерно одинаковое количество данных, так что при увеличении протяжённости во времени размер тайла в пространстве уменьшается.*

Решаем мы эту проблему хранением сразу нескольких наборов тайлов различной формы. Условно говоря, есть тайлы за каждый отдельный день, тайлы за неделю, за месяц и за год. Список тайлов передаётся на клиент. Когда пользователь выбирает определённый период в календарике, наш API решает, какой набор данных лучше использовать и грузит нужные тайлы.

Процесс построения мультивременных тайлов выглядит следующим образом. Сначала мы формируем тайлы

длительностью в, скажем, один день. Берём исходные данные, и выбираем из них данные за первый день (мы предполагаем, что с каждым объектом связана определённая дата). По отобранным данным строим тайлы как для обыкновенного векторного слоя (в тайлах, конечно, запоминаем, что это тайлы за первый день). Затем выбираем данные за второй день, строим для них векторные тайлы и т.п. пока не переберём все дни, за которые есть данные. На следующем этапе начинаем формировать тайлы протяжённостью в, например, неделю. Выбираем из исходных данных все объекты, относящиеся к первой неделе и строим для них тайлы как для простого векторного слоя. Затем берём данные за вторую неделю и т.п. Далее мы можем аналогично построить наборы тайлов за месяц, квартал, год и т.п. Описание всех полученных тайлов передаётся на клиент.

На клиенте при показе на карте мультिवременного слоя API вычисляет временные рамки запроса и выбирает подмножество тайлов, которые нужно запросить с сервера. Пример: пусть на сервере хранятся тайлы длительностью день и неделя, а пользователь выбрал период в 10 дней. У нас есть вариант запросить тайлы за 10 дней по отдельности, за неделю и отдельно за 3 оставшихся дня, или за две недели. Последний вариант может оказаться предпочтительнее, если общее количество данных мало (из-за меньшего числа запросов на сервер).

Поскольку все тайлы имеют примерно одинаковый размер, основным критерием выбора в ГеоМиксере является общее количество запросов на сервер для получения всех нужных данных. Если протяжённости во времени всех тайлов кратны друг другу (скажем, 1, 4, 16, 64 дня), минимизировать эту величину можно за один проход по списку тайлов (например, при помощи какой-нибудь древовидной структуры).

Описанное решение позволяет крайне эффективно с точки зрения сервера выбирать данные для запросов разной формы (фактически, раздаётся статика). Из минусов можно отметить увеличение объёма данных на сервере для хранения векторных тайлов (хотя для нас это всё-равно копейки по сравнению с размером наших растровых данных) и сложность обновления (каждое изменение нужно отразить сразу в нескольких тайлах).

## Банальное заключение

Мультिवременные слои в ГеоМиксере — мощный инструмент визуализации и анализа данных, содержащих временную компоненту. Сочетание эффективной выборки данных за любой промежуток времени с богатыми возможностями визуализации позволяет строить на основе мультिवременных данных очень интересные проекты. Минимальная нагрузка на сервер позволяет хранить данные практически неограниченного размера.

## (Бес)полезные ссылки

- [Краткий обзор GeoMixer WEB-GIS](#)
- [Пользовательская документация GeoMixer](#)
- [GeoMixer API Reference](#)

[Обсудить в форуме](#) Комментариев — 14

Последнее обновление: 2014-05-15 02:08

Дата создания:

Автор(ы): [parshin](#)