NATIONAL RESEARCH UNIVERSITY

АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

Пространственные структуры данных
Spatial Data Structures

Рамон Антонио Родригес Залепинос
arodriges@hse.ru

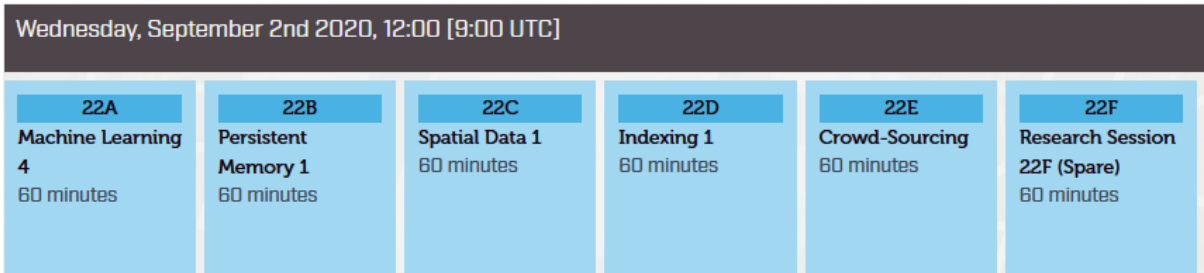
Структура модуля 2

Модуль № 2	№	Дата	Тема лекции	№	Домашние задания
7	1	27 окт	Хеширование, хэш таблицы 2	2b	ДЗ-Каникулы
8	2	03 ноя	Фильтры	3	Задание на C++
9	3	10 ноя	СД для вторичной памяти	4	Задание на C++
10	4	17 ноя	Пространственные СД	5	Задание на C++
11	5	24 ноя	Параллельные СД	6	Задание на C++ и/или C# (зависит от выбранного уровня сложности)
12	6	01 дек	Параллельные СД 2		
13	?	08 дек	Деревья в оперативной памяти		Примерно за 2 недели заканчиваются ДЗ
14	8	15 дек	Современные тренды		
		СЕССИЯ с 21.12.2020			

Highlights:

- лекции, семинары и ДЗ синхронизированы
- ? – возможно будет еще одна лекция, с другой темой
- некоторые представления об эффективности СД развеяны
  - на семинаре вы собственноручно, на практике сравните производительность красно-черных деревьев и хэш таблиц (если не нужны next & prev)
- пространственные СД – обширный класс: **помимо int & string ∃ и другие типы данных**
  - в современном мире, **около 80% всех данных содержат географическую привязку**: ссылка 1 (Forbes), ссылка 2 (Carto)
  - отдельные секции на значимых конференциях (e.g., VLDB: https://vldb2020.org/program.html)

Требование к студентам на лекции: слушайте внимательно!



GDAL

https://github.com/OSGeo/gdal

- Одна из самых популярных библиотек работы с геопространтсвенными данными
- 1,448,133 строк кода (октябрь 2020)

rouault Merge pull request #3180 from rouault/vrt_preserve_block_size ... ✕ 0b7879c 3 hours ago 43,257 commits

.github	code_checks.yml: run flake8 on swig/python/osgeo/utils and swig/pytho...	16 days ago
autotest	Merge pull request #3180 from rouault/vrt_preserve_block_size	3 hours ago
gdal	Merge pull request #3180 from rouault/vrt_preserve_block_size	3 hours ago
.editorconfig	Disable default text=auto .gitattributes	2 years ago
.gitattributes	Disable default text=auto .gitattributes	2 years ago
.gitignore	MRF: add support for interleaved LERC (#2846)	3 months ago
.travis.yml	Revert ".travis.yml: disable s390x target since Travis-CI infra is KO...	5 months ago
CITATION	CITATION: update to 2020 [ci skip]	9 months ago
CONTRIBUTING.md	Fix typo in CONTRIBUTING.md	14 months ago
README.md	README.md: remove table with tested configurations [ci skip]	6 months ago
Vagrantfile	Vagrant: new config key for LXC driver	2 years ago
appveyor.yml	CI: use http instead of https for download.osgeo.org due to bad SSL c...	last month

README.md

GDAL - Geospatial Data Abstraction Library

GDAL is an open source X/MIT licensed translator library for raster and vector geospatial data formats.

gdal.org

raster geospatial-data vector

Readme

Releases 94

GDAL 3.2.0 Latest
14 days ago

+ 93 releases

Packages

No packages published

Contributors 259

План лекции

Часть I. Пространственные данные (spatial data) – **короткий обзор**

Что это? Почему важно? Какие особенности? Где узнать больше?

1. Мотивация: разнообразие типов данных
2. Актуальность & ценность данных (6V)
3. Квалификация для работы с данными: пришел, увидел, победил?
   - пример задачи с R-деревом (пространственная СД)
   - Geoapplications Development (курс Разработка Геоприложений)

Часть II. Структура данных «Классическое R-дерево» (1984 г.)

≈ B-tree для spatial data ⇒ рассматриваем R-tree после B-tree

1. «Постановка задачи»
2. Идеи устройства R-tree
3. Поиск и вставка в R-tree
4. Разбиение узлов R-tree
5. Удаление из R-tree

Часть III. Резюме лекции

ДЗ по пространственным СД:
на 8 баллов («отлично»)
не нужно реализовывать R-tree

Часть I

Мотивация

- **Помимо int & string ∃ и другие типы данных**
- До этой лекции мы рассматривали чистые строки и числа, но не всегда приходится работать только со строками и числами
- Есть типы данных, характеристики которых кардинально отличаются от строк и чисел
- пространственные СД – обширный класс
  - в современном мире, **около 80% всех данных содержат географическую привязку**: ссылка 1 (Forbes), ссылка 2 (Carto)
  - отдельные секции на значимых конференциях (e.g., VLDB: https://vldb2020.org/program.html)

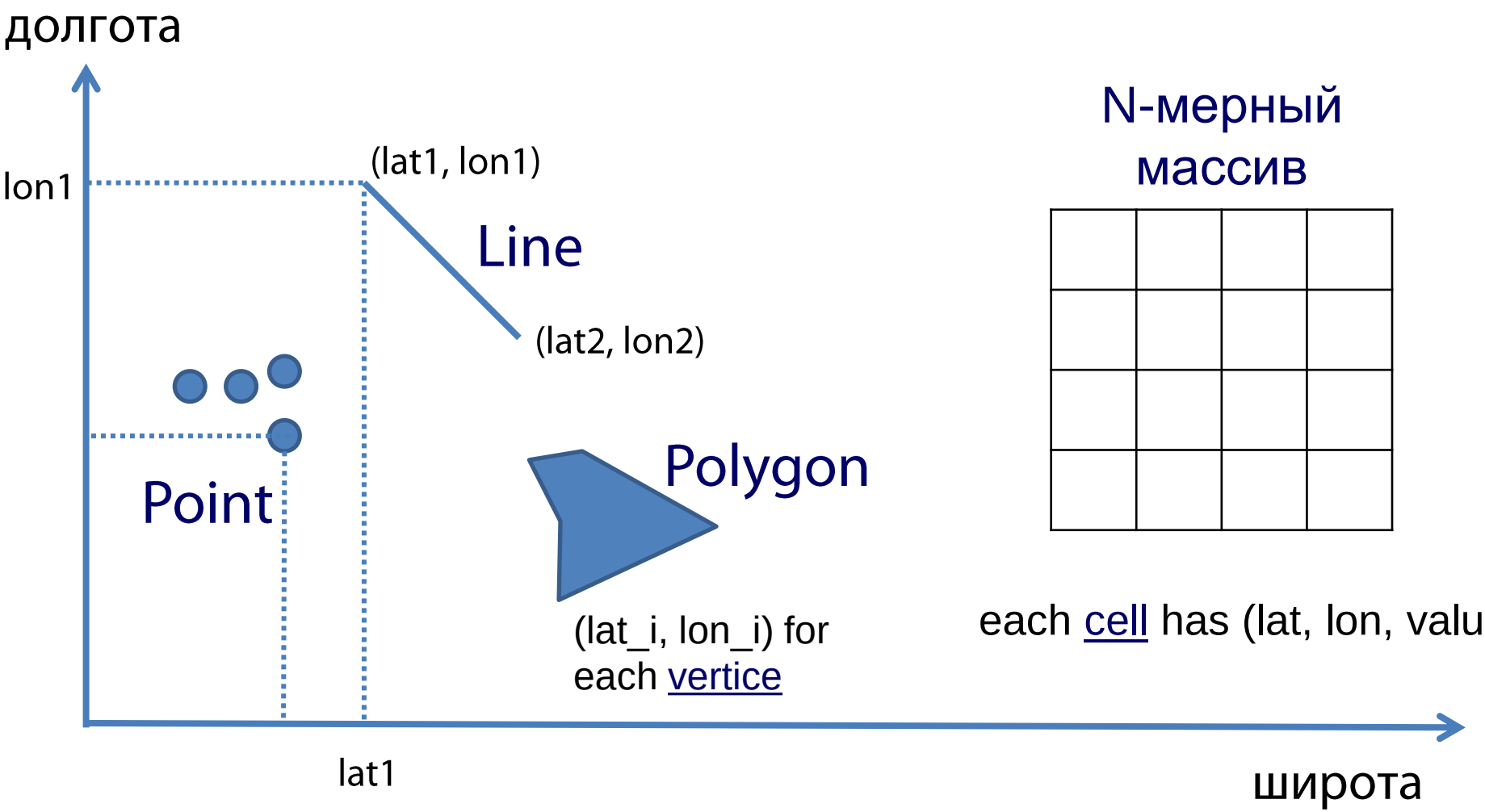
Wednesday, September 2nd 2020, 12:00 [9:00 UTC]

22A	22B	22C	22D	22E	22F
Machine Learning 4 60 minutes	Persistent Memory 1 60 minutes	Spatial Data 1 60 minutes	Indexing 1 60 minutes	Crowd-Sourcing 60 minutes	Research Session 22F (Spare) 60 minutes

Два основных типа геоданных

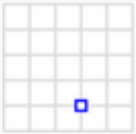
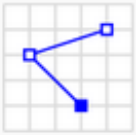
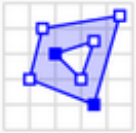
ВЕКТОРНЫЕ

РАСТРОВЫЕ

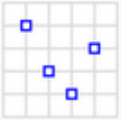
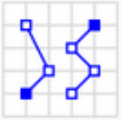
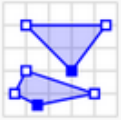


Векторные данные

Geometry primitives (2D)

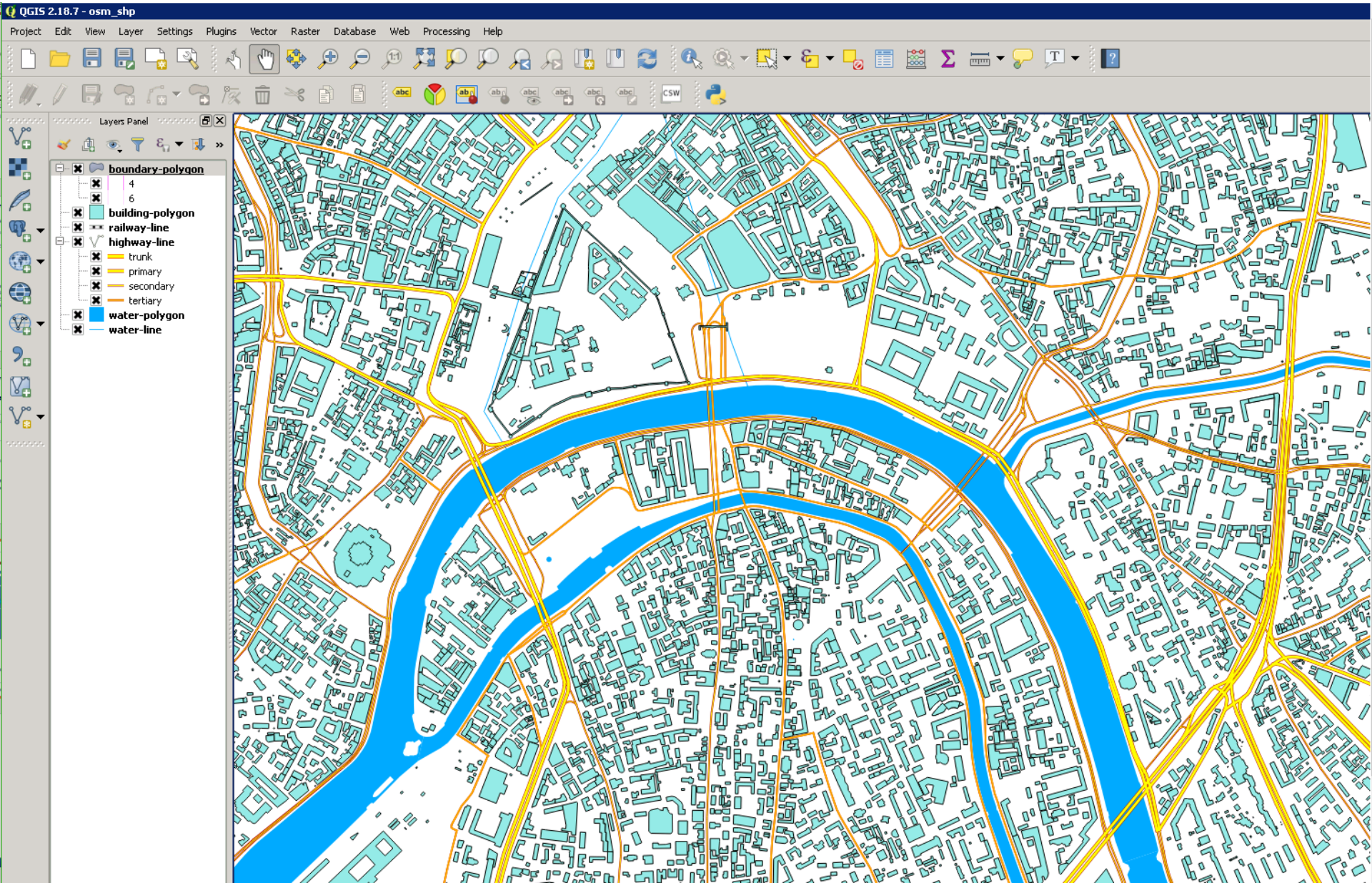
Type	Examples	
Point		POINT (30 10)
LineString		LINESTRING (30 10, 10 30, 40 40)
Polygon		POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))
		POLYGON ((35 10, 45 45, 15 40, 10 20, 35 10), (20 30, 35 35, 30 20, 20 30))

Multipart geometries (2D)

Type	Examples	
MultiPoint		MULTIPOINT ((10 40), (40 30), (20 20), (30 10))
		MULTIPOINT (10 40, 40 30, 20 20, 30 10)
MultiLineString		MULTILINESTRING ((10 10, 20 20, 10 40), (40 40, 30 30, 40 20, 30 10))
MultiPolygon		MULTIPOLYGON (((30 20, 45 40, 10 40, 30 20)), ((15 5, 40 10, 10 20, 5 10, 15 5)))
		MULTIPOLYGON (((40 40, 20 45, 45 30, 40 40)), ((20 35, 10 30, 10 10, 30 5, 45 20, 20 35), (30 20, 20 15, 20 25, 30 20)))

https://en.wikipedia.org/wiki/Well-known_text

Пример векторных данных: Москва



Растровые данные: спутник 30 см/пиксель

15 см/пиксель – после обработки

http://microsites.digitalglobe.com/30cm/

Актуальность

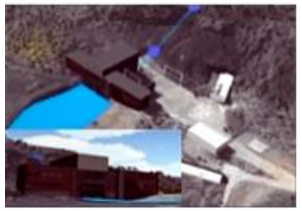
Большие данные

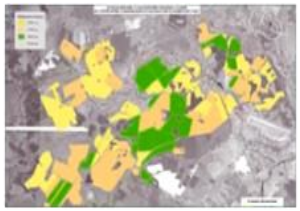
- ECMWF data warehouse: **137.5 млн. файлов, ~52,7 ПБ** (2015)
- DigitalGlobe, провайдер спутн. данных: **~80 ТБ/день** (2017)
- Planet**:** «селфи Земли каждый день», **400 спутников** (2021)

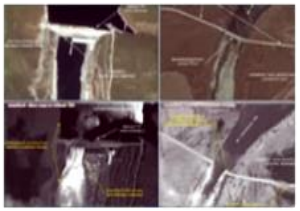
Коммерческая ценность

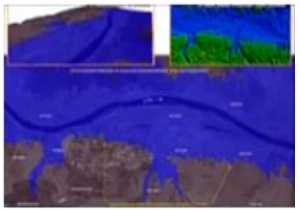
UBER MasterCard IBM GeoHack CROC

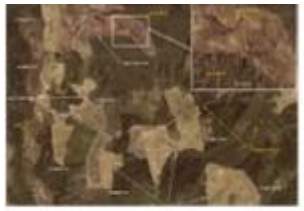
Практическая польза











6Vs: Пространственные данные – большие данные?

VOLUME	VARIETY	VELOCITY	VERACITY	VALUE	VARIABILITY
The amount of data from myriad sources.	The types of data: structured, semi-structured, unstructured.	The speed at which big data is generated.	The degree to which big data can be trusted.	The business value of the data collected.	The ways in which the big data can be used and formatted.

https://www.quora.com/What-are-the-six-Vs-of-Big-Data

Volume & Velocity: Maxar (бывший DigitalGlobe)

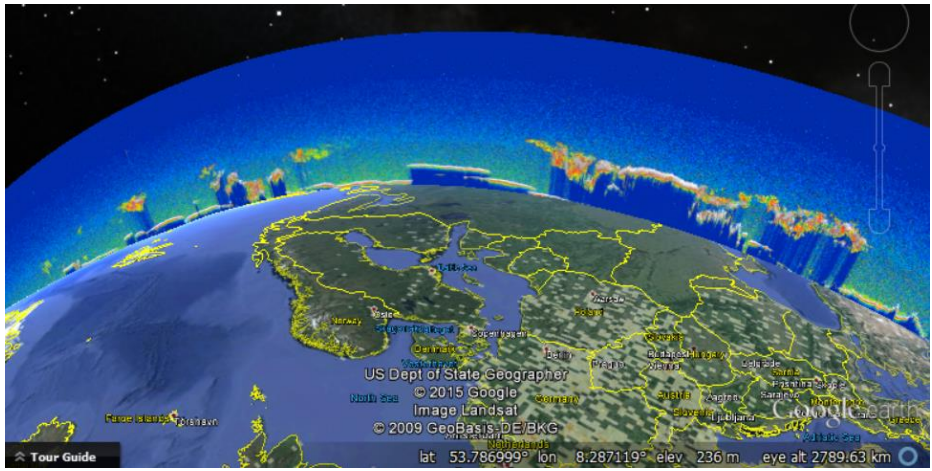
- **Собирает 80 терабайт в день**
- **Накопил в облаке Amazon более 100 петабайт**
- Это только лишь один из операторов ДЗЗ

https://youtu.be/mkKkSRIxU8M

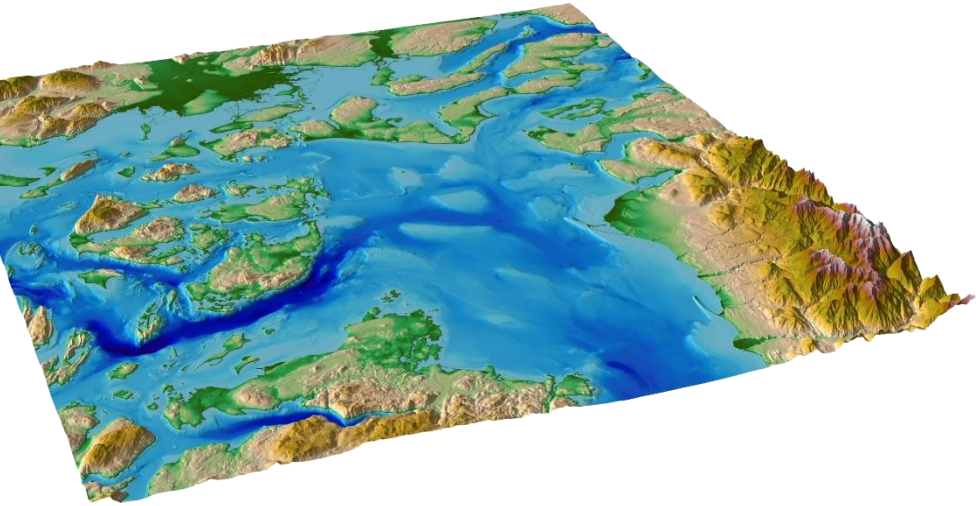
Variety

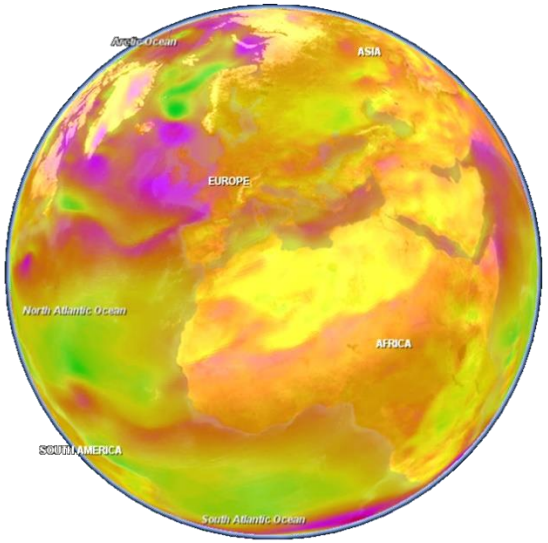
- **1D, 2D, 3D, 4D, …**
- **Более 140 форматов**
- **Разнообразные структуры**

Вертикальный профиль аэрозоля (CALIPSO)



Цифровая модель рельефа



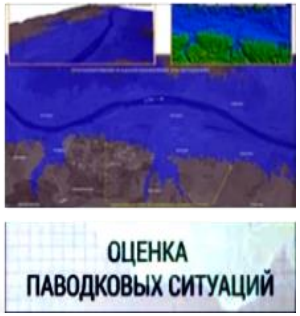


Скорость ветра (ассимиляция, модель)

Value (польза) & variability (варианты использования)

С помощью растровых геоданных можно отвечать на такие вопросы:

- **Чрезвычайные ситуации**: какие районы города будут затоплены при повышении уровня воды на 1 метр (используются DEM)?
- **Сельское хозяйство**: как нужно удобрять поле, чтобы собрать больше урожая (точное земледелие, precision agriculture)?
- **Городское планирование**: где и какие санитарно-гигиенические мероприятия нужно проводить, чтобы повысить качество воздуха?



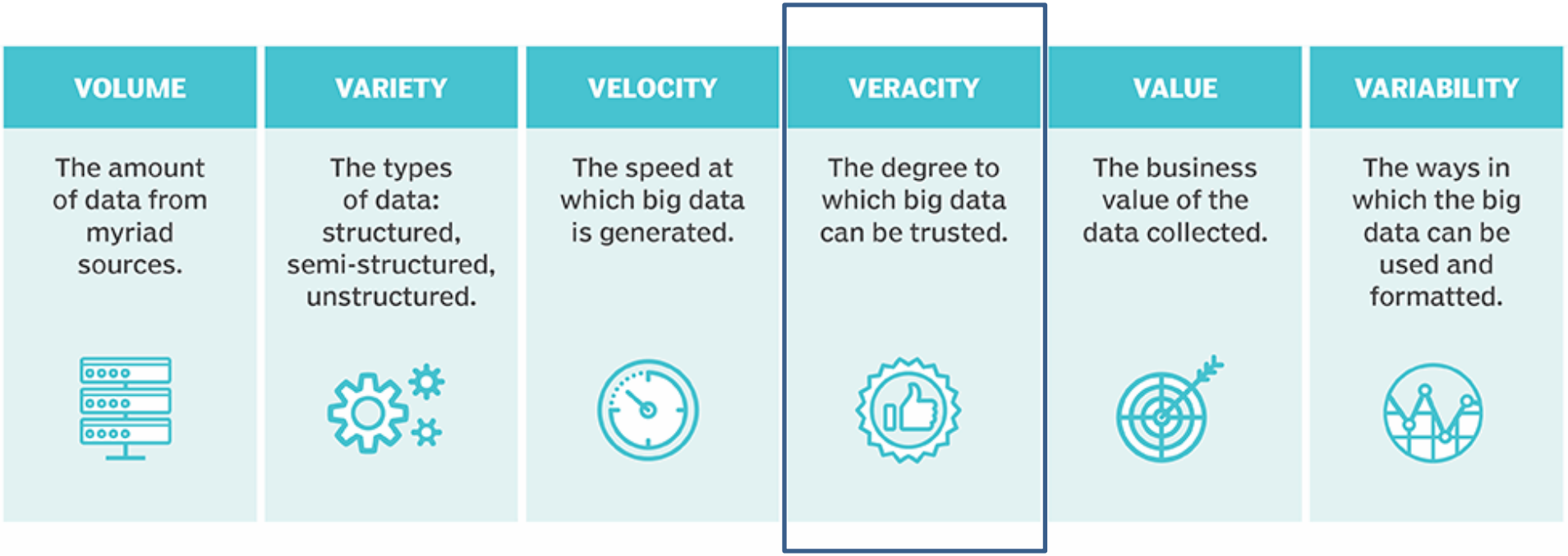
https://learn.arcgis.com/ru/arcgis-imagery-book/

Роскосмос "Цифровая Земля"

https://www.youtube.com/watch?v=LsR_gfCeSlQ

Veracity – достоверность

- **Это физические измерения**
- **Федеральный закон 2017 г. (лесной кодекс):** арендатор лесного участка обязан прикладывать к отчету об использовании леса материалы данных дистанционного зондирования Земли, подтверждающие соблюдение границ выделенного участка под вырубку
- **Площадь лесов России – около 8,09 млн. км² (20% от всех лесов мира)**



Для решения задач с пространственными данными

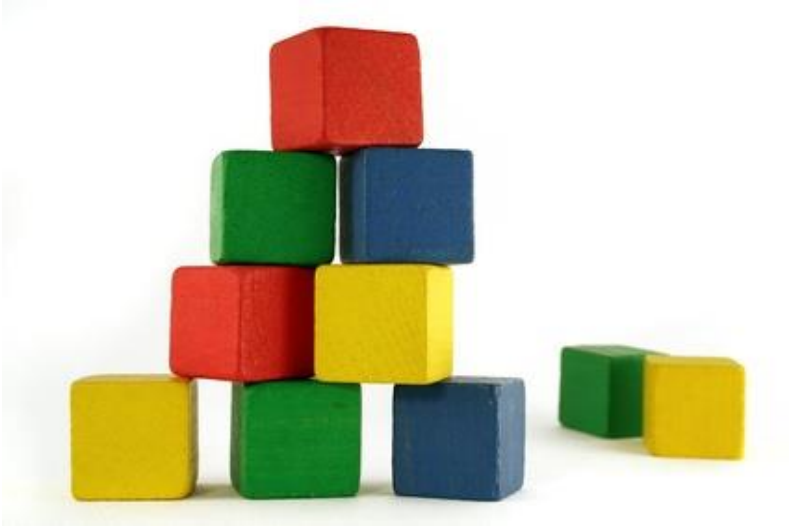
Нужна особая квалификация

в т.ч. особые структуры данных

Нужно владеть

особыми технологиями и **особой** теорией –

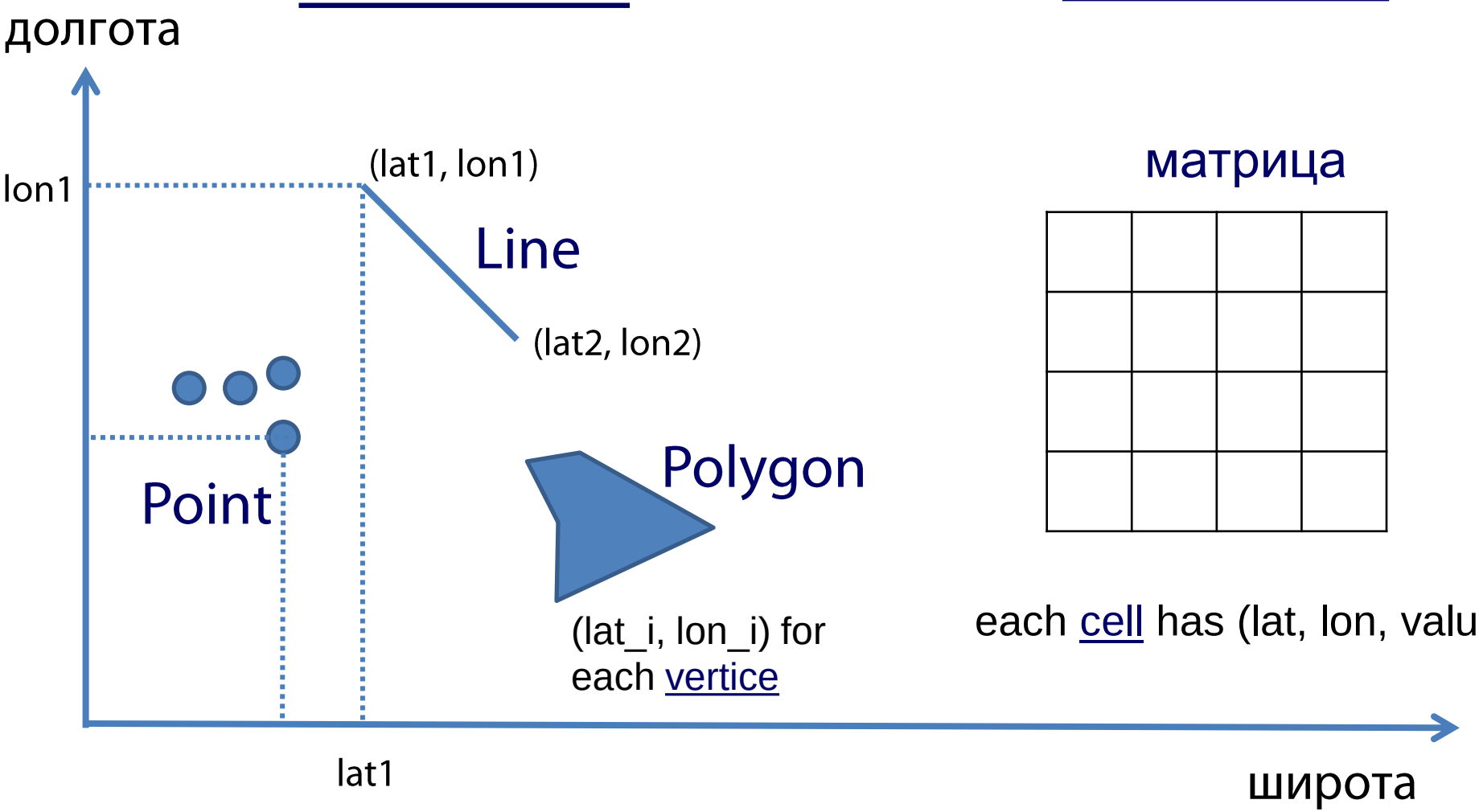
строительные блоки для решения реальных задач



Два основных типа геоданных

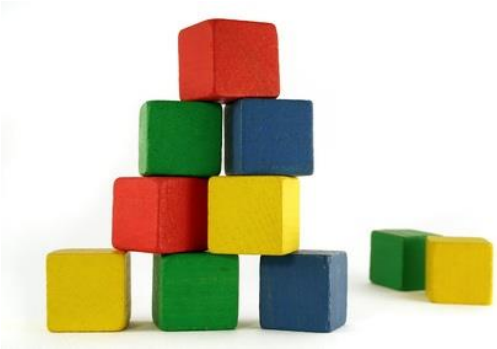
ВЕКТОРНЫЕ

РАСТРОВЫЕ



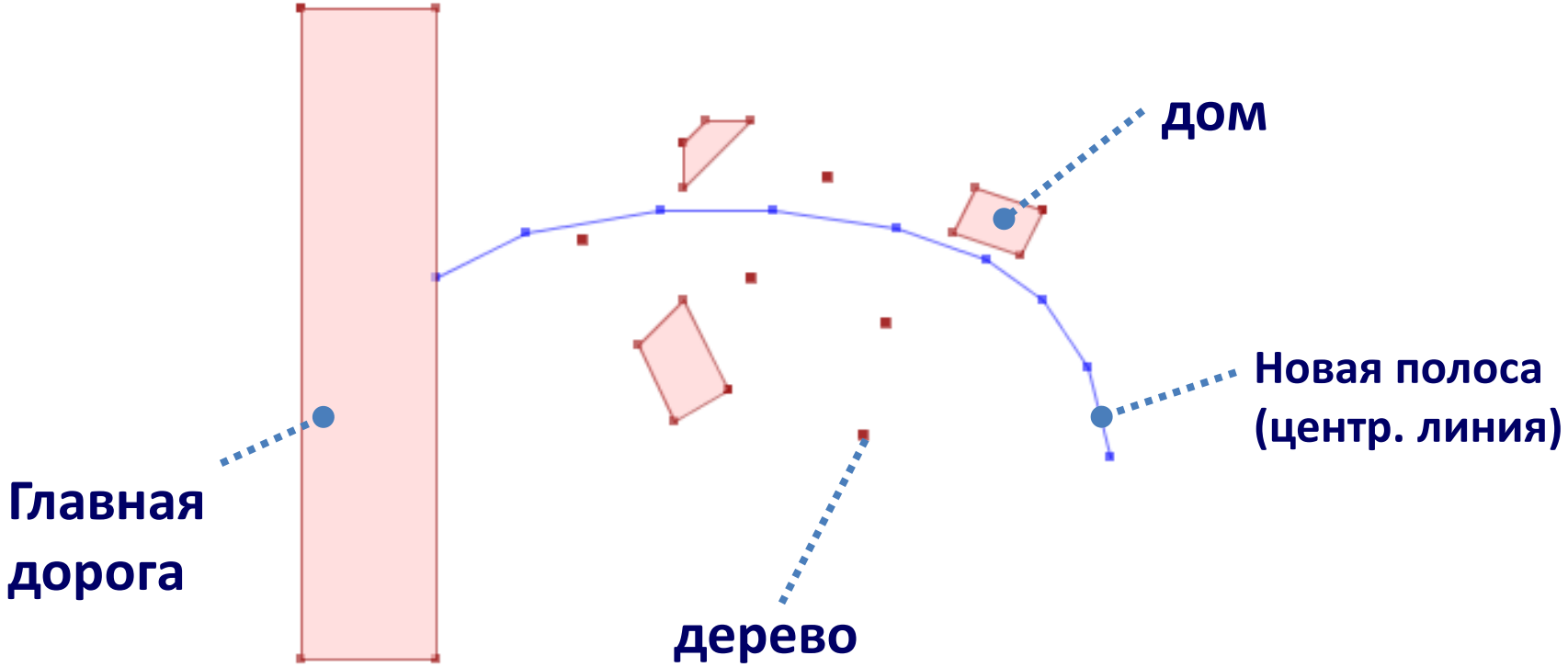
Фундаментальная разница от других типов данных: координаты в пространстве

Особые теория и технологии – строительные блоки для решения реальных задач



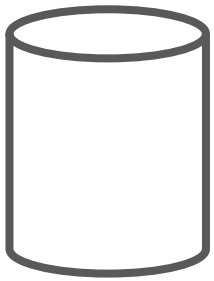
Пример

Ввиду увеличения трафика городские власти хотят расширить главную дорогу и добавить вторую полосу движения шириной 20 метров. Найти объекты, которые должны быть снесены.



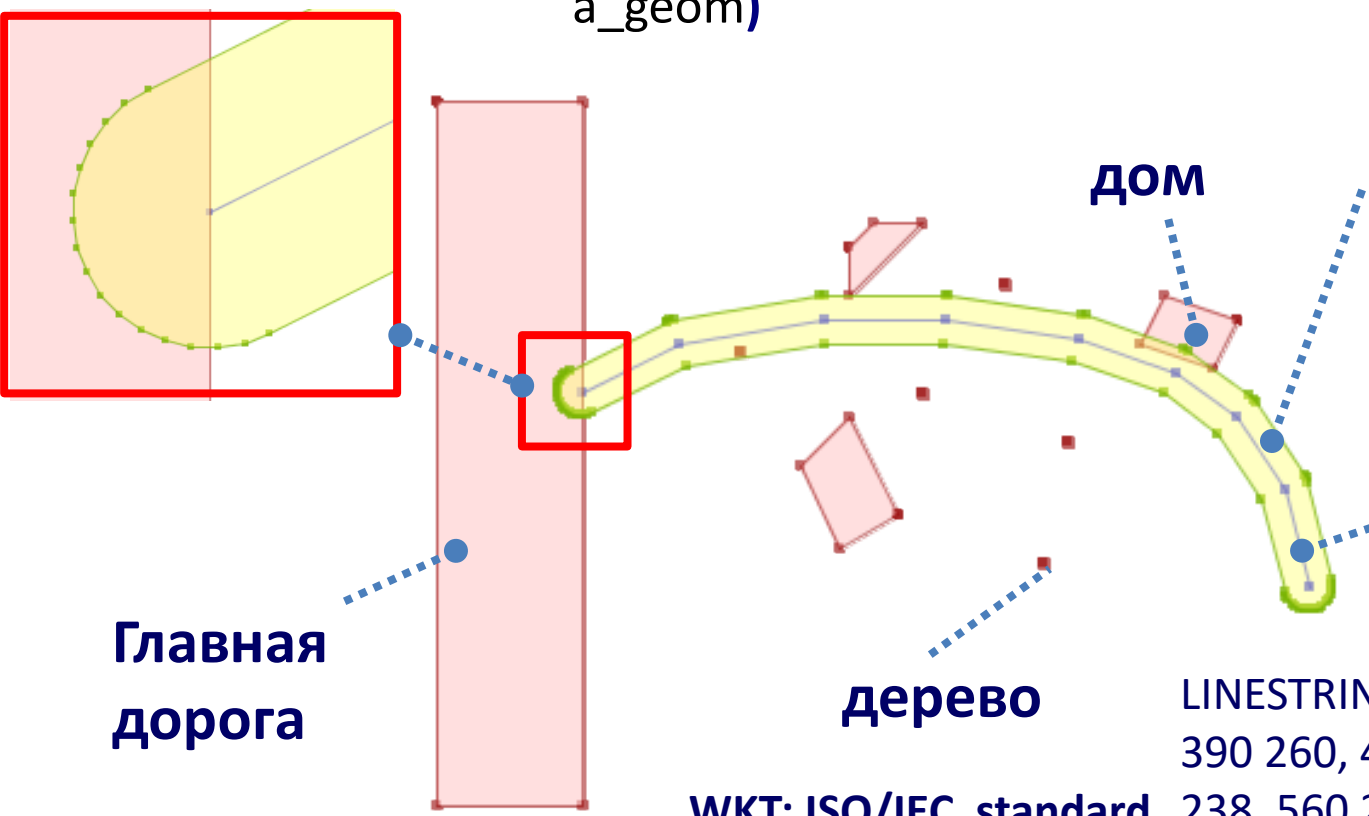
Решение (СУБД, WKT, Buffer, Топология, SQL, …)

СУБД



```
SELECT a_geom, address, owner
FROM city_database
WHERE ST_Intersects (
        ST_Buffer(
            ST_GeomFromText("LINESTRING (290 230,…) ",4326),
            10),
        a_geom)
```

R-дерево – пространственная структура данных
(избегаем полного перебора)



BUFFER (ST_Buffer)

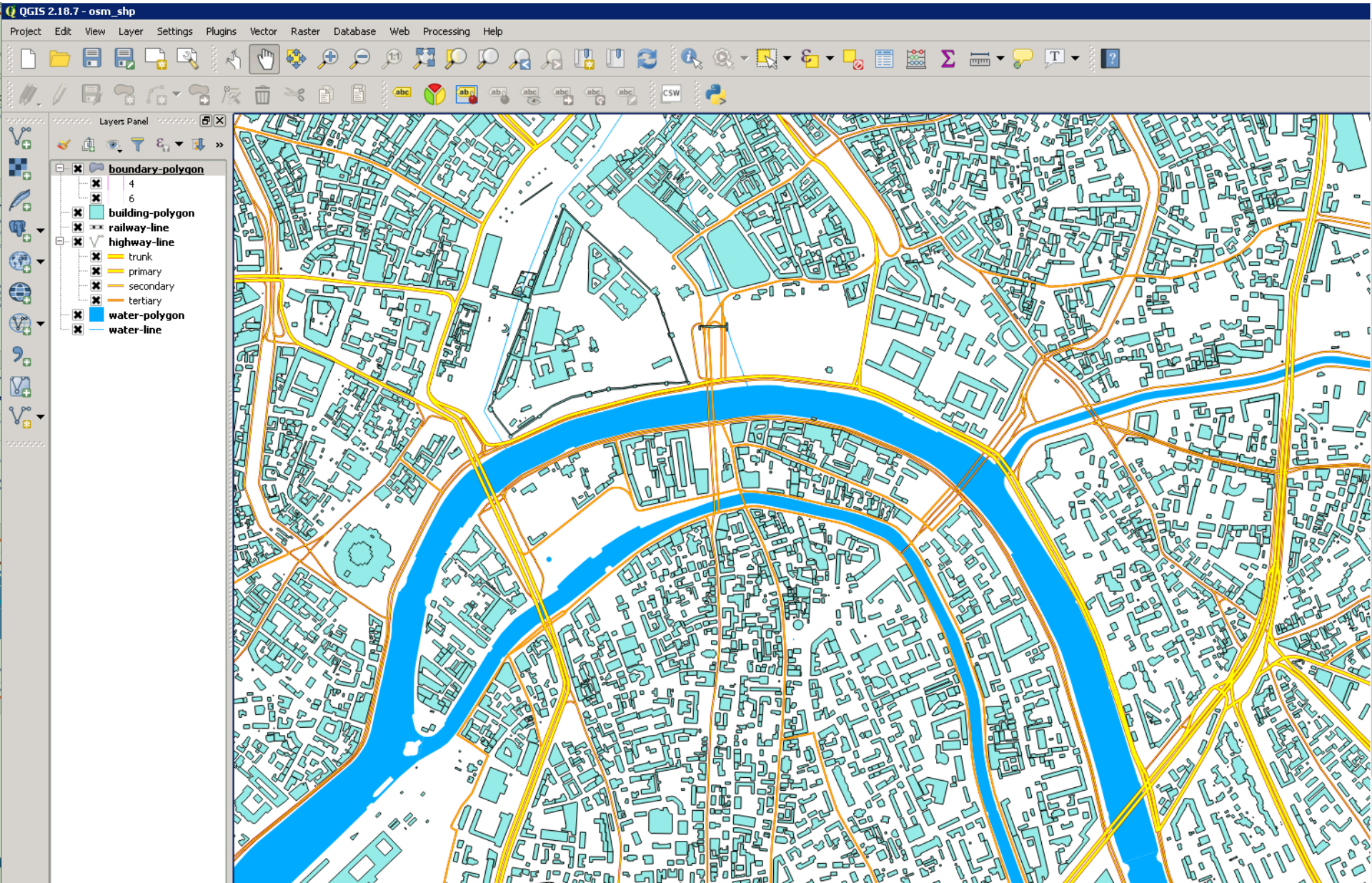
POLYGON ((325.5278640450004 258.94427190999915, 326.90755828109235, ….))

Новая полоса (центр. линия)

LINESTRING (290 230, 330 250, 390 260, 440 260, 495 252, 535 238, 560 220, 580 190, 590 150)

WKT: ISO/IEC standard

Пример векторных данных: Москва



Получить квалификацию:

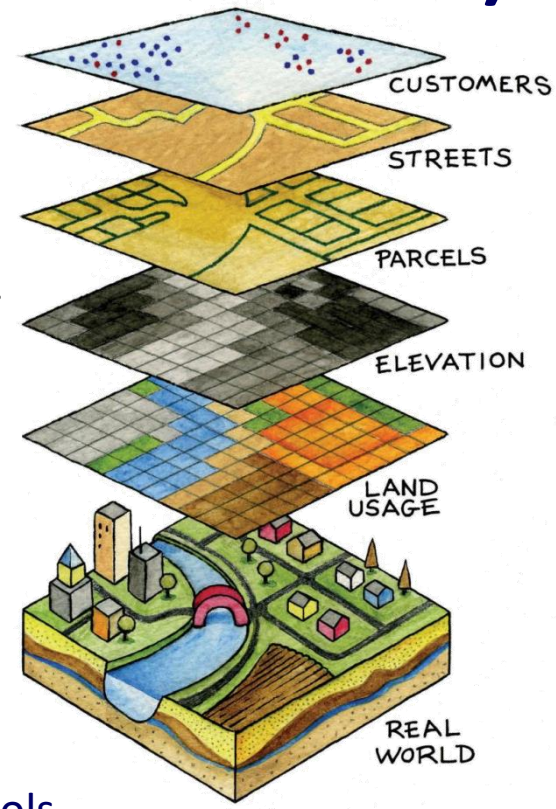
Предмет по выбору, 3 курс

Разработка геоприложений
Geoapplications development

Полный стек теории и технологий работы с геопространственными данными

(c) R.A. Rodriges Zalipynis

Geospatial technology stack (not exhaustive)



Visualization

CARTODB Geospatial on the cloud

MapBox

Leaflet

Processing

Libraries in
Java, Python, R, JavaScript, …

Transmission

Netty

network protocols
WCS, WFS, WMS, OPeNDAP

Storage

DBMS

PostGIS Spatial PostgreSQL

Almost all DBMS are geo-enabled

H2GIS

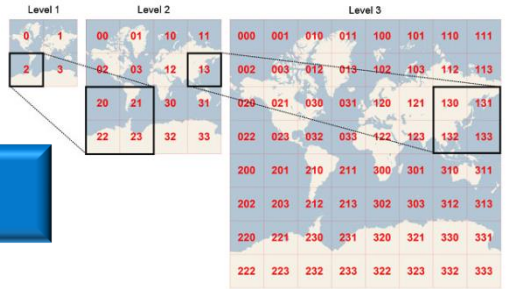
Files

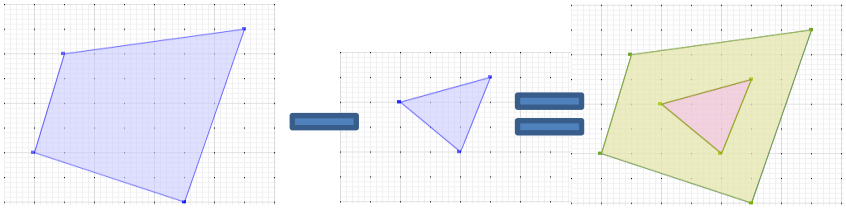
SHP, KML, GeoJSON, GeoTIFF, NetCDF, HDF

Geospatial theory stack (not exhaustive)

GPS, WiFi, geohash, …

Processing





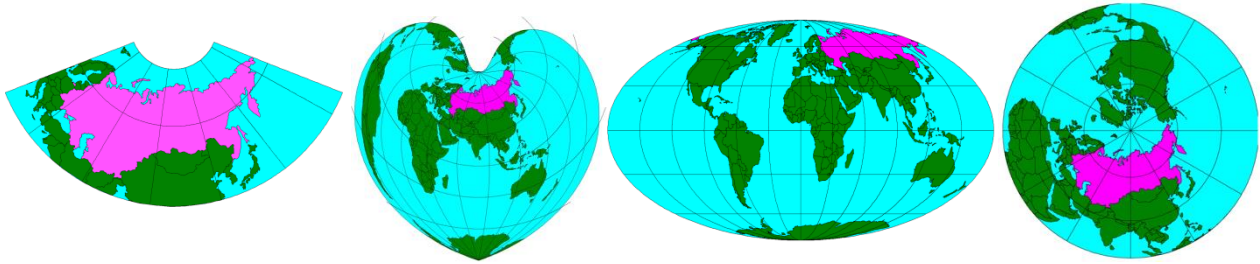
Topology

Notations

```
MULTILINESTRING ((10 10, 20 20, 10 40),
(40 40, 30 30, 40 20, 30 10))
```

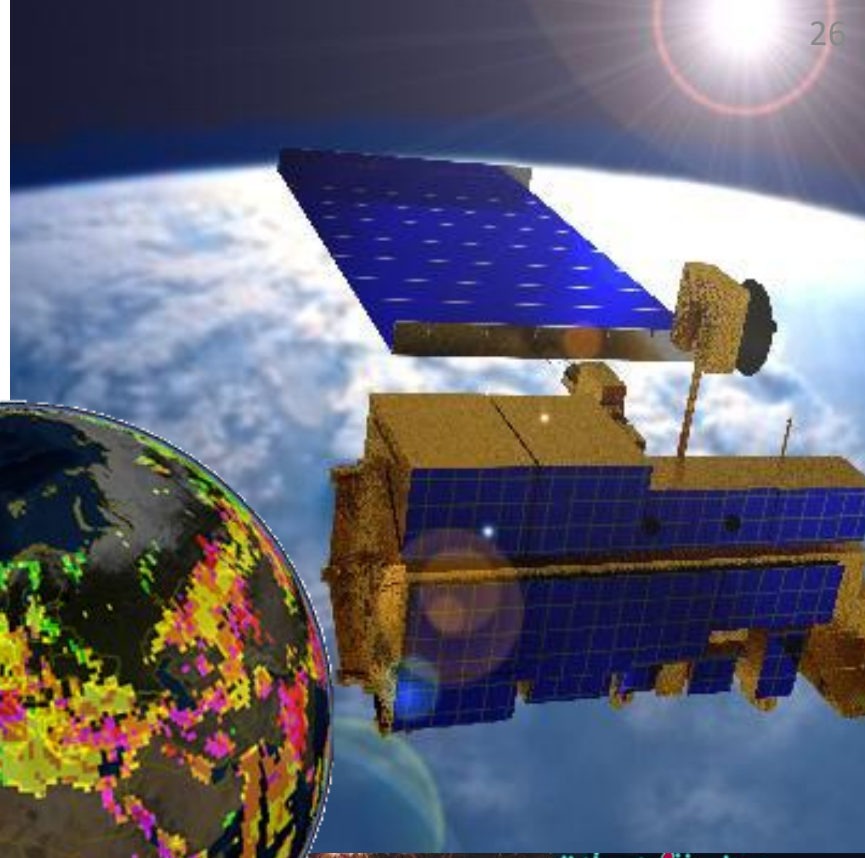
ISO/IEC standard

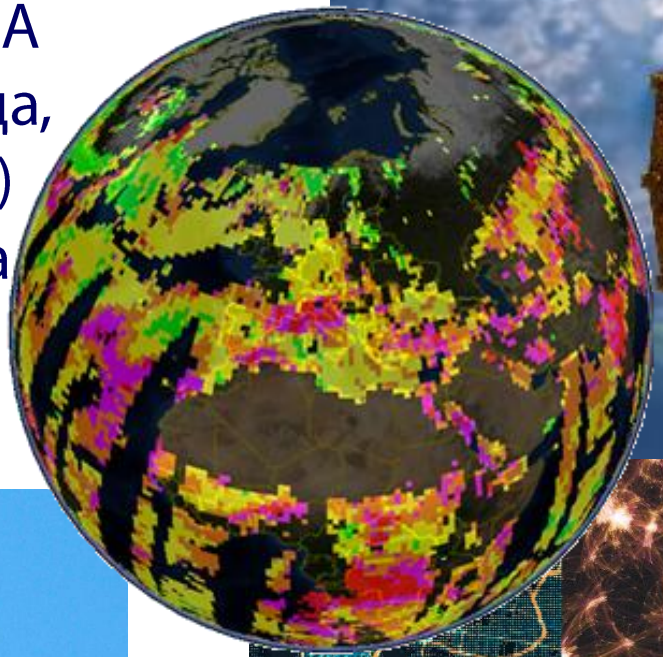
Projections

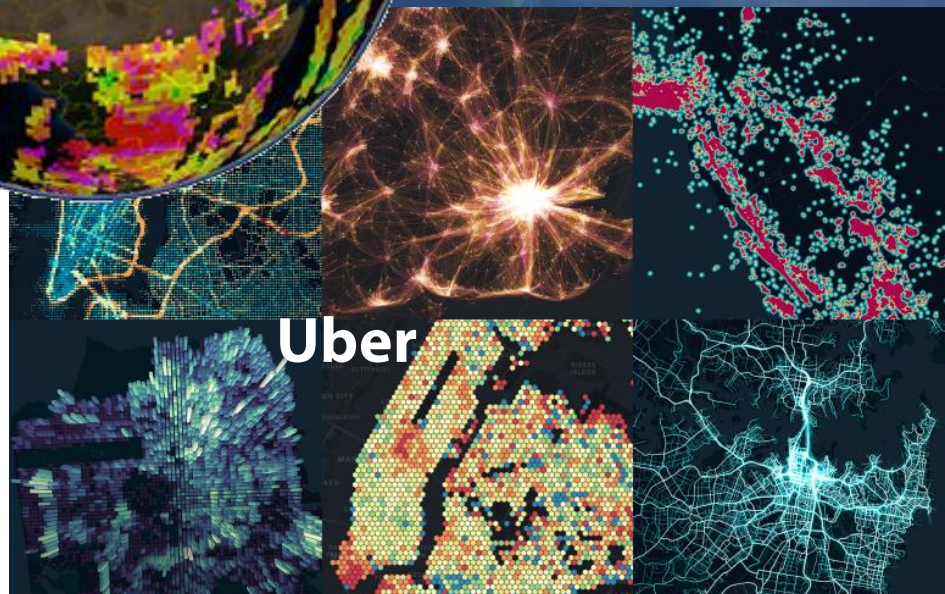


Геоинформатика: хранение, обработка, передача, анализ, визуализация данных с географической привязкой

- Спутники, лидары, БПЛА
- Моделирование (погода, климат, гидрология, …)
- Мобильные устройства
- Любые данные с геопривязкой



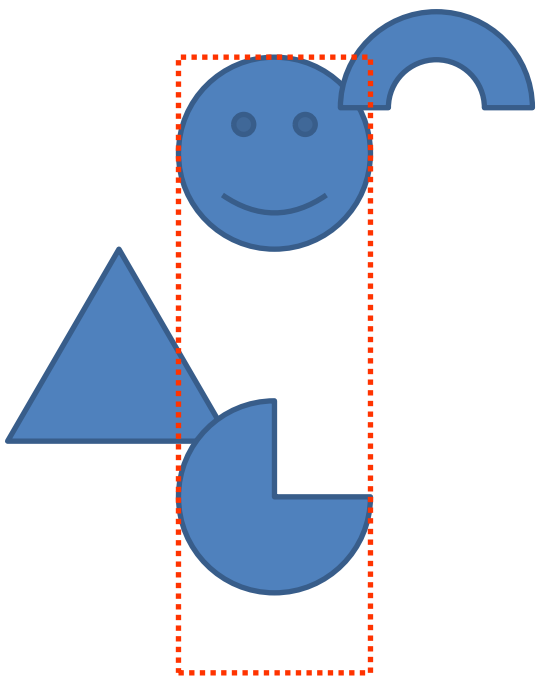




Часть II

Постановка задачи: range-queries

Вход: (1) пространственные объекты $\{g_1, g_2, \dots, g_N\}$
(2) область (прямоугольник, окружность, полигон, …)
Выход: пространственные объекты $\{g_i\}$ внутри заданной области



Подход 1
(straightforward)**:**

Вычислить пересечение каждого объекта с областью – много объектов, операция сложная

Подход 2:
Построить структуру данных, выполнить быструю фильтрацию, протестировать на пересечение оставшиеся объекты

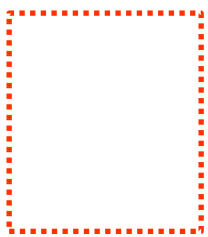
Структура данных R-Tree

Для эффективного выполнения range-queries

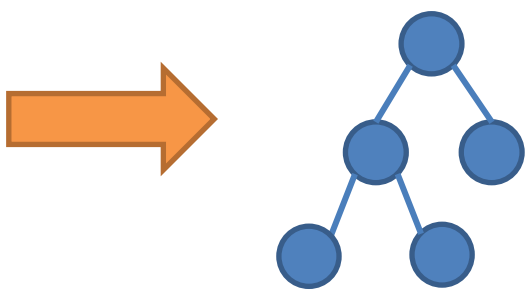
Для векторных данных

Antonin Guttman. *R-Trees: A Dynamic Index Structure for Spatial Searching*, Proc. 1984 ACM **SIGMOD** International Conference on Management of Data, pp. 47–57.

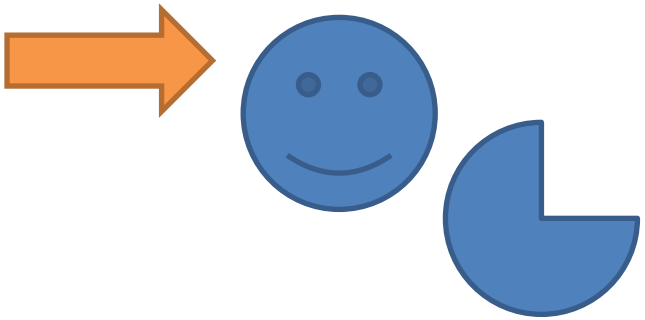
Вход:
прямоугольная область



R-Tree

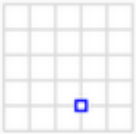
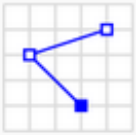
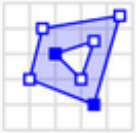


Выход:
Объекты (точнее их MBR), которые пересекаются с прямоугольной областью

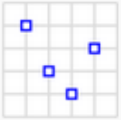
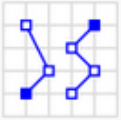
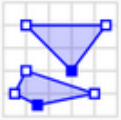


Векторные данные

Geometry primitives (2D)

Type	Examples	
Point		POINT (30 10)
LineString		LINESTRING (30 10, 10 30, 40 40)
Polygon		POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))
		POLYGON ((35 10, 45 45, 15 40, 10 20, 35 10), (20 30, 35 35, 30 20, 20 30))

Multipart geometries (2D)

Type	Examples	
MultiPoint		MULTIPOINT ((10 40), (40 30), (20 20), (30 10))
		MULTIPOINT (10 40, 40 30, 20 20, 30 10)
MultiLineString		MULTILINESTRING ((10 10, 20 20, 10 40), (40 40, 30 30, 40 20, 30 10))
MultiPolygon		MULTIPOLYGON (((30 20, 45 40, 10 40, 30 20)), ((15 5, 40 10, 10 20, 5 10, 15 5)))
		MULTIPOLYGON (((40 40, 20 45, 45 30, 40 40)), ((20 35, 10 30, 10 10, 30 5, 45 20, 20 35), (30 20, 20 15, 20 25, 30 20)))

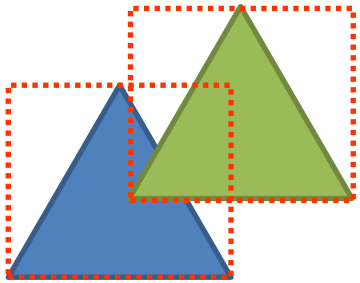
https://en.wikipedia.org/wiki/Well-known_text

MBR (Minimal Bounding Rectangle)

Минимальный ограничивающий прямоугольник –

(1) минимальный по размеру прямоугольник, в который вписывается пространственный объект

(2) стороны || осям координат

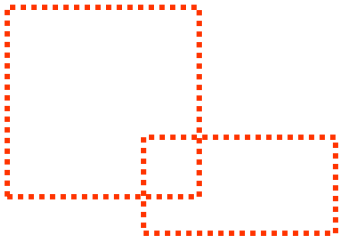


Пересечение

(а) геометрия-геометрия: сложно и долго

(б) MBR-MBR: просто и быстро

```
boolean intersects(other) {
 return !(other.minx > maxx ||
    other.maxx < minx ||
    other.miny > maxy ||
    other.maxy < miny);
}
```



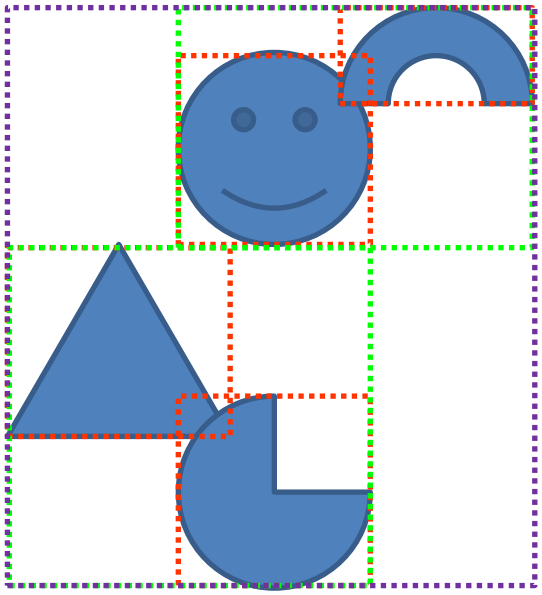
https://en.wikipedia.org/wiki/Minimum_bounding_box
https://en.wikipedia.org/wiki/Minimum_bounding_rectangle

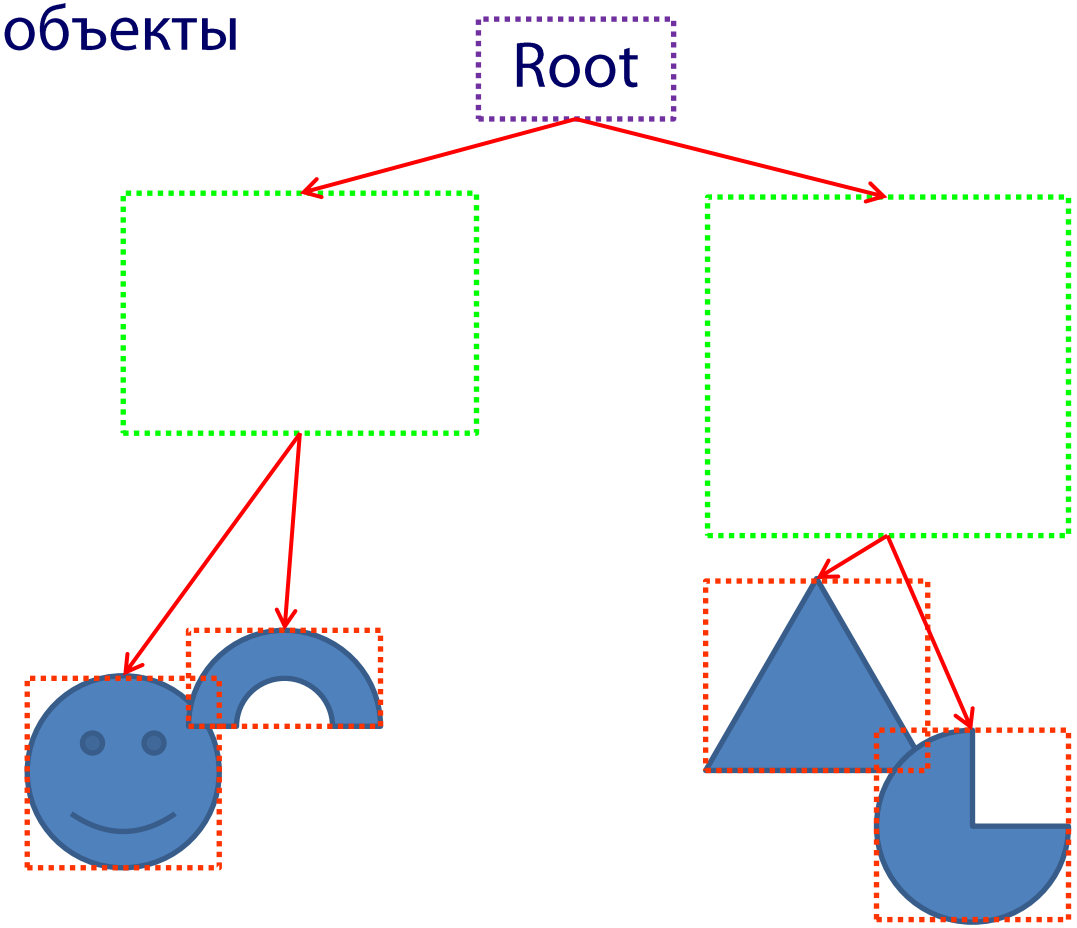
Структура данных R-Tree

- Antonin Guttman,1984
- R-Tree = Rectangle (Region)-Tree
- Похоже на B-Tree (балансировка)
- Представить каждый объект MBR
- Группировать соседние объекты

В узлах: MBR поддеревьев

В листьях: MBR и указатели на объекты





Свойства R-Tree

Наследство от B-tree

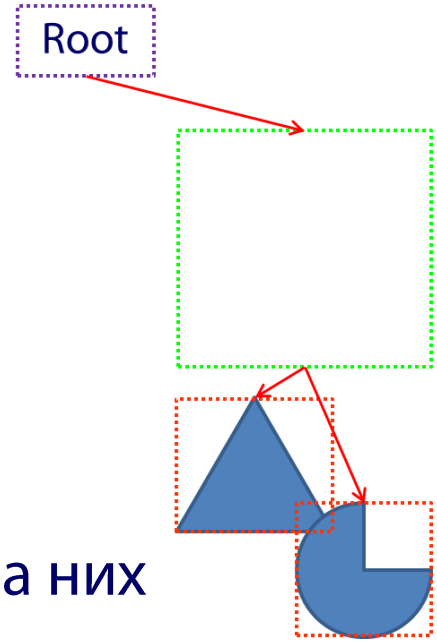
- Создано для вторичной памяти (читаем/пишем «по многу»)
- Каждый лист содержит от m до M (B-дерево: только параметр t)
  - ссылок на объекты (если это не корень)
  - MBR
- Все листья на одной высоте
- Каждый узел (кроме корня) содержит от m до M записей
- Высота дерева $O(\log_m N)$, где N – число объектов

Наследство от B+tree

- Только листья содержат ссылки на объекты

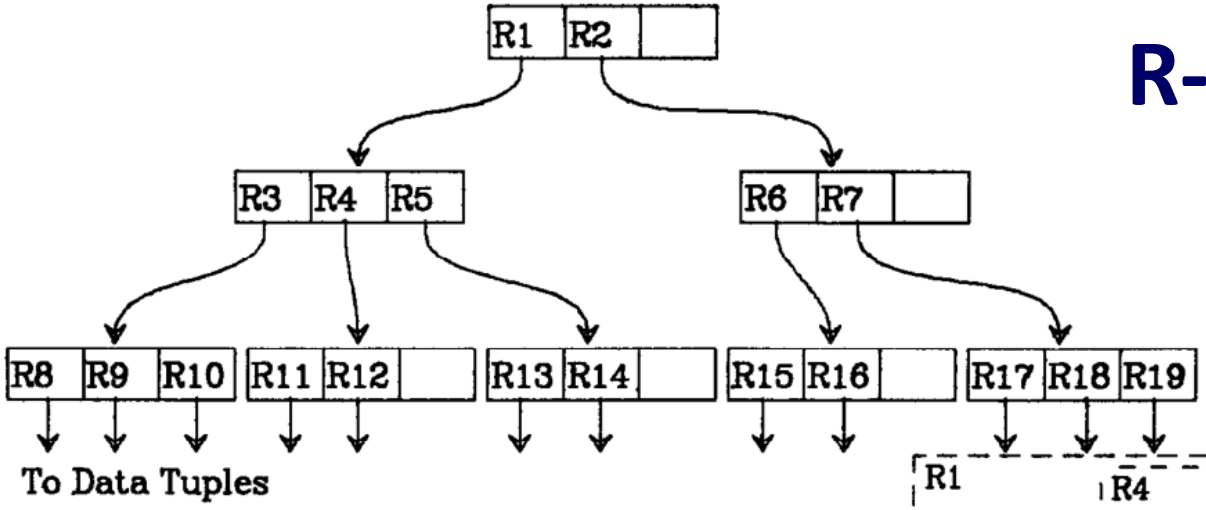
Новое для R-tree

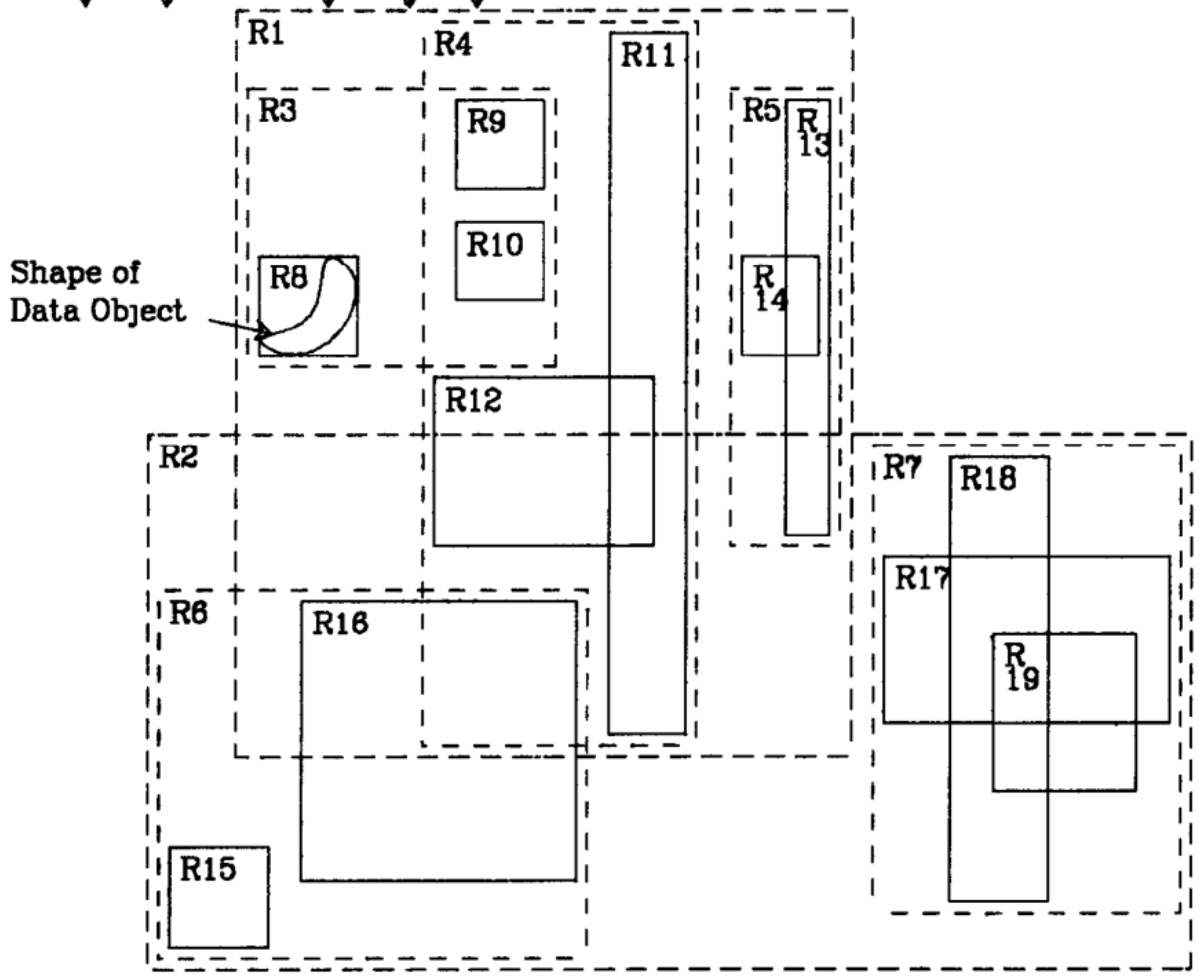
- Узел (не лист) содержит
  - Общий MBR для всех поддеревьев
  - MBR_i и $child_i$ – MBR поддеревьев и ссылки на них



R-Tree – пример

Дерево и возможные перекрытия [1]





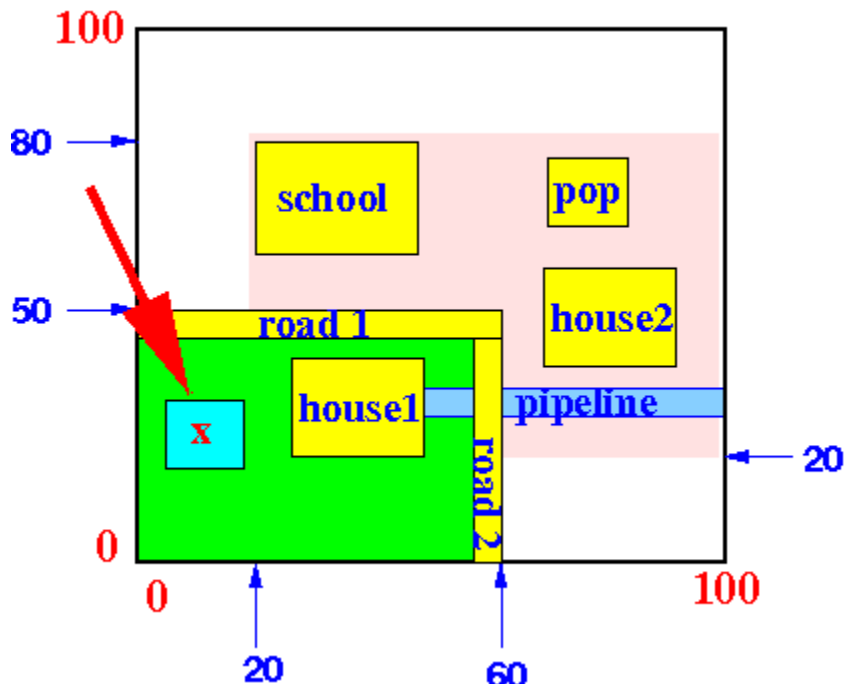
Один объект может покрываться/ пересекаться с несколькими узлами одновременно

Алгоритм поиска?

R-Tree – вставка (случай 1)

- Алгоритм похож на вставку в B-tree
- Нужно найти лист, куда вставим MBR объекта
- По пути к листу мы не модифицируем узлы в отличие от B-tree

Случай 1: существует узел с MBR, который полностью покроет MBR нового объекта



link

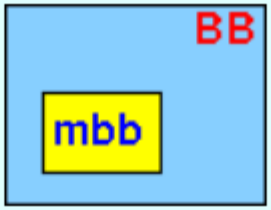
Зайти в этот узел и продолжить рекурсивно

Если таких узлов несколько, выбрать узел с наименьшим MBR по площади

R-Tree – вставка

(случай 1)

```
/* ==================================================
   Insert( (mbb,ObjID), n)

       (mbb,ObjID) = object with given mbb
       n           = current node of operation
   ================================================== */
Insert( (mbb, ObjID), n )
{
   // n = current node of the search in the R-tree

   if ( n == internal node )
   {
      for ( each entry ( BB,  childptr ) in node n ) do
      {
         /* ==========================================================
            Insert object in subtree if its mbb is contained by BB
            ==========================================================
         if ( mbb ⊆ BB )
         {
```



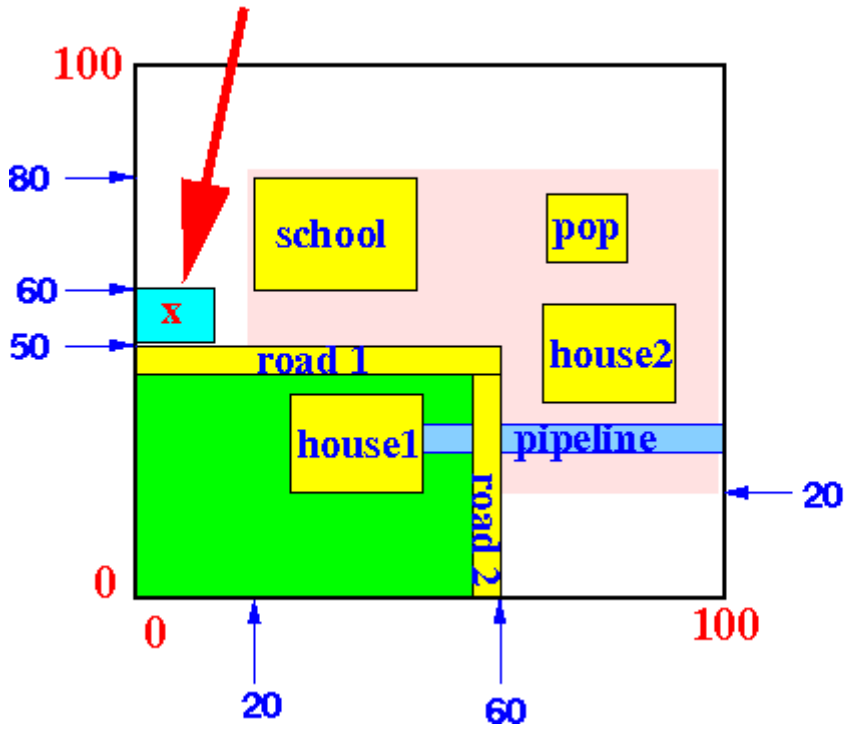

```
            /* ---------------------------------------------
               Insert the object in this subtree !!!
               --------------------------------------------- */
            Insert( (mbb,ObjID),  childptr);   // Just recurse !!

            /* ---------------------------------------------
               The recursion WILL have inserted the object !
               --------------------------------------------- */
            return;            // Terminates the recursion !!!
         }
      }
```

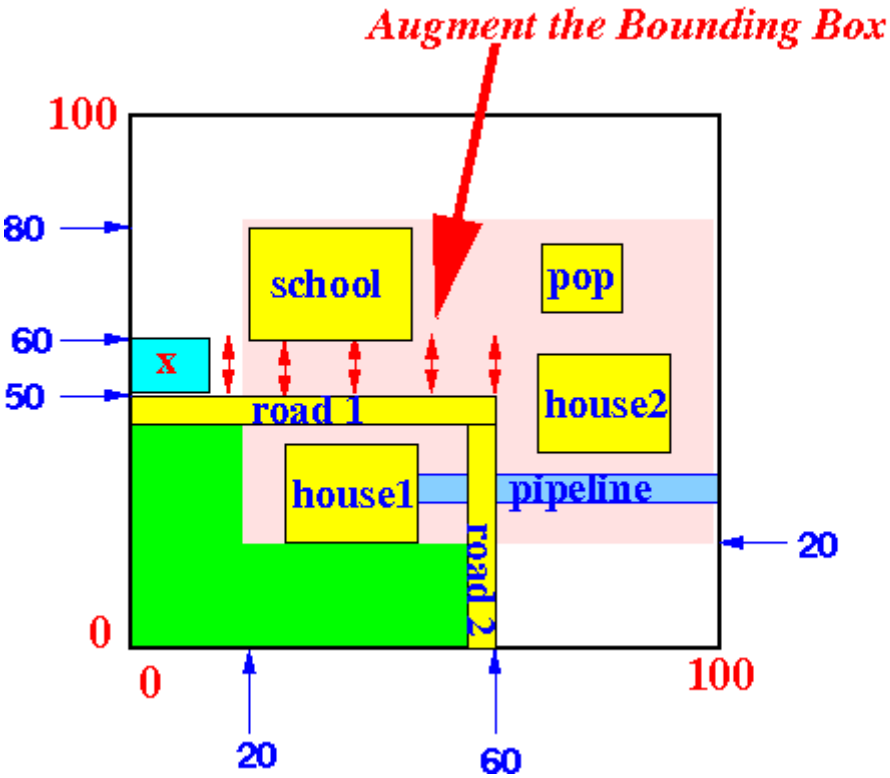
В этом коде не выбирается узел с наименьшим MBR по площади, но первый попавшийся: для упрощения кода

link

R-Tree – вставка (случай 2)

- **Случай 2**: нужно расширить MBR узла
- Выбирается такой MBR, чтобы его меньше всего расширять
- Если таких несколько, выбрать с наименьшей площадью
- В отличие от B-tree мы **не** разбиваем заранее все узлы по пути от корня до листа, но поднимаемся вверх при необходимости



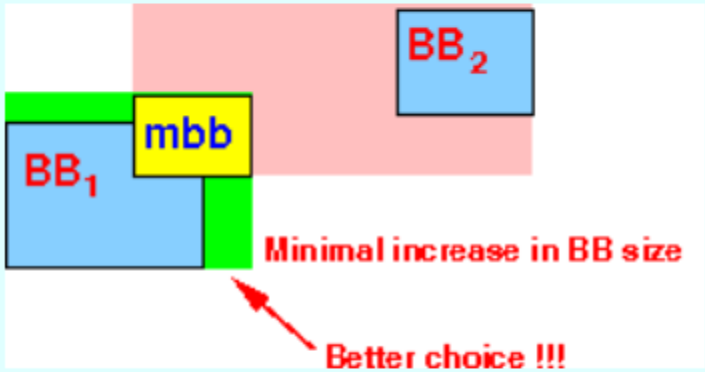


link

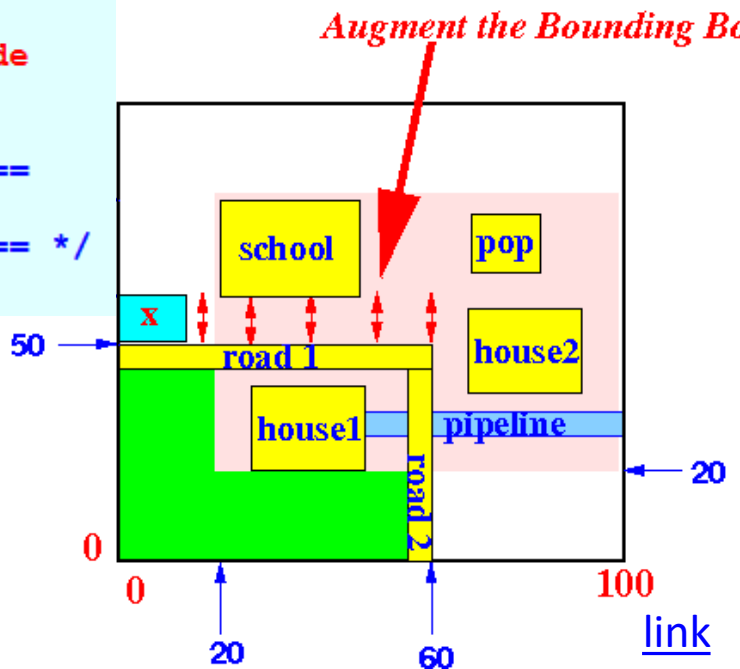
R-Tree – вставка

(случай 2)

```
/* ****************************************************
   We can ONLY reach here when EVERY BB in the node n
   FAILS to contain the object !!!
   ***************************************************** */
Find a BB in n such that:
    Enlarging BB to contain the object will add
    the least amount of area to the bounding box BB
```




```
Update this BB in n by the Enlarged BB;

(Note: We do not need to update the BB in the parent node
because THAT BB contains the insert object)

/* ====================================================
   Now the enlarged BB contains the object !!!
   ==================================================== */
Insert( (mbb,ObjID),  childptr);   // Just recurse !!
```



link

R-Tree – вставка

(завершение, вставка в лист)

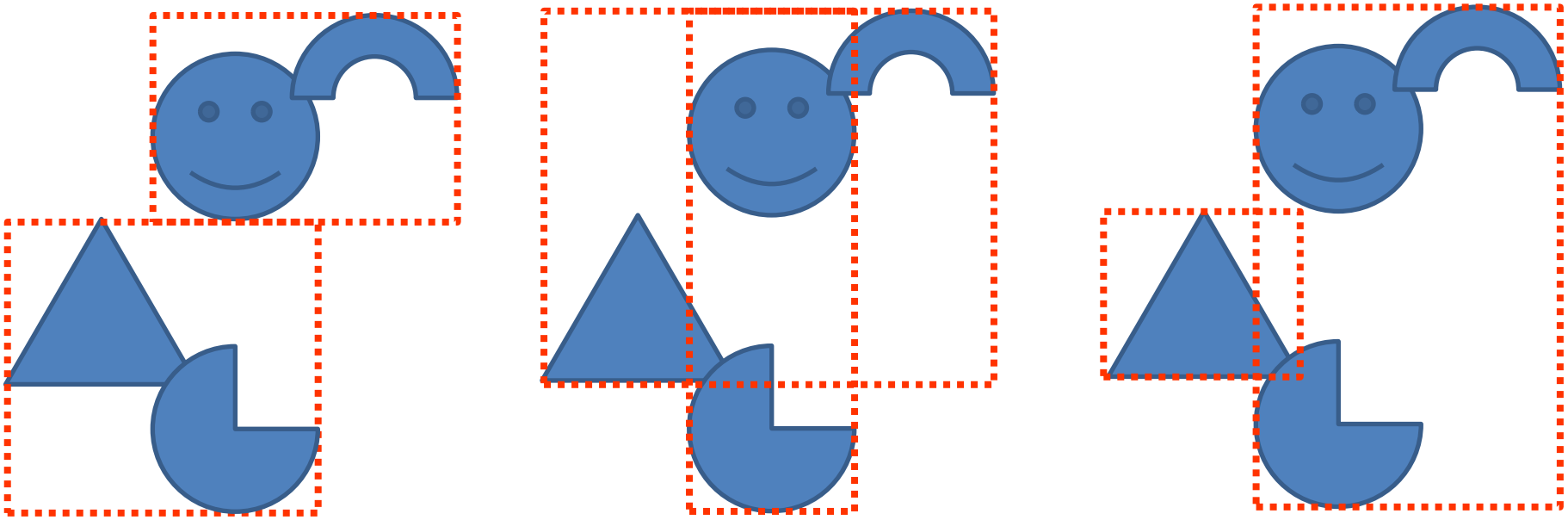
```
else
{
   /* ==================================================
      We found the leaf node to hold the object !!!
      ================================================== */
   if ( leaf node has space to hold object )
   {
      insert (mbb, ObjID) in the leaf node;
   }
   else
   {
      Split the leaf node into 2 nodes;  // Tough...

      Find the bounding box for all the objects in each node;

      Replace the parent's (1) BB by of the 2 bounding boxes;

      (The parent node can overflow and split.
       You need a Insert procedure into an internal node
       - just like the B-tree algorithm)

   }
}
```

Переполнение листа, может произойти как и в B-дереве

link

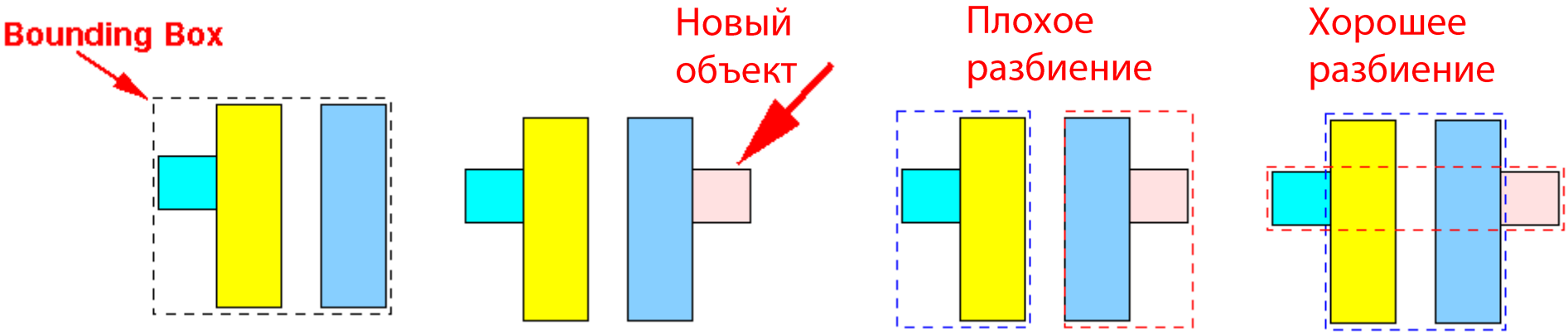
R-Tree: лист переполнен

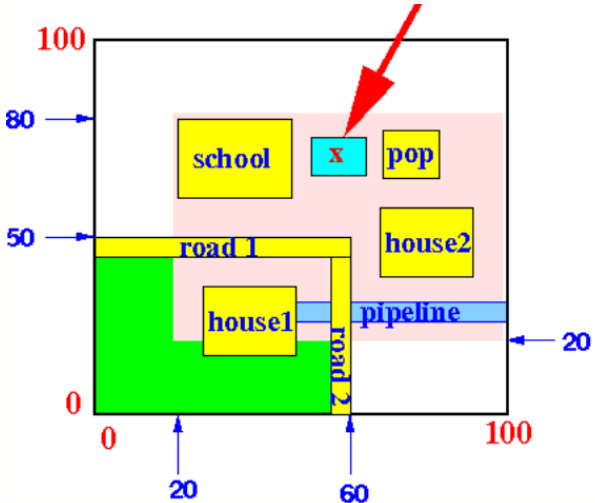
- Есть много способов объединять объекты
- И много способов разбивать переполненный лист
- И многие из этих способов не очень хорошие

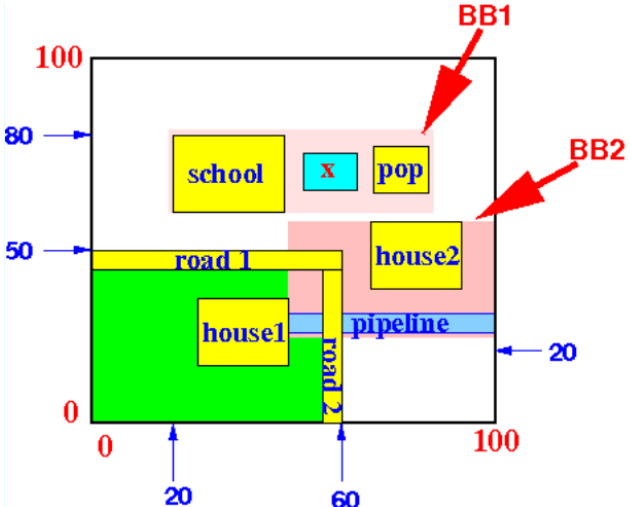


R-Tree: как разбивать лист?

- Нужно создавать такие MBR при разбиении, чтобы как можно меньшее их количество исследовалось при поиске объекта
- Идея-предположение (интуитивно): чем меньше MBR, тем он точнее описывает свои объекты
- Очевидный и не очевидный способы







link

R-Tree: как разбивать лист – много подходов

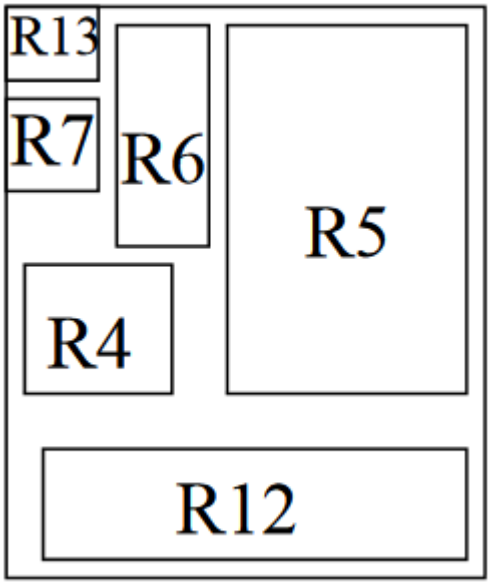
Если переполняется родительский узел, применяем к нему такую же тактику

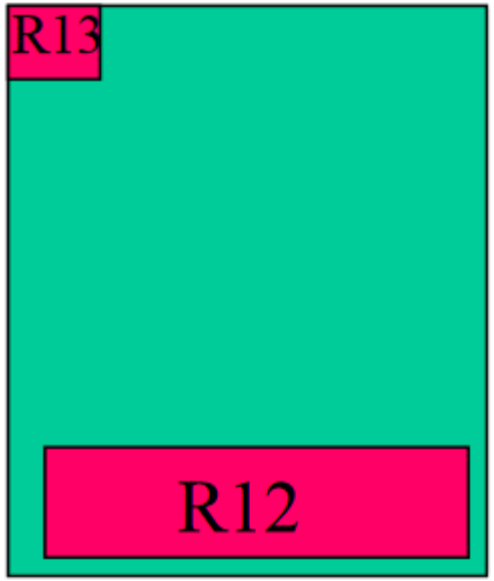
- **Подход 1.** Полный перебор всех возможных разбиений, выбор двух минимальных MBR

Все остальные подходы **эвристические** – работают быстрее полного перебора, находят хорошее, но не лучшее решение

R-Tree: как разбивать лист – $O(M^2)$

- Выбрать опорные объекты E_1 и E_2: d – максимально
$$d = area(MBR(E_1, E_2)) - area(E_1) - area(E_2)$$
- У нас две группы, представленные E_1 и E_2
- Если в одной из групп объектов $< m$ и для того, чтобы ее размер стал как минимум m нужно все оставшиеся объекты назначить этой группе – то ∀ оставшиеся put в эту группу
- Добавить объект к группе, MBR которой менее всего расширяется; несколько таких групп – выбрать группу с наименьшим числом объектов; иначе – выбрать любую группу



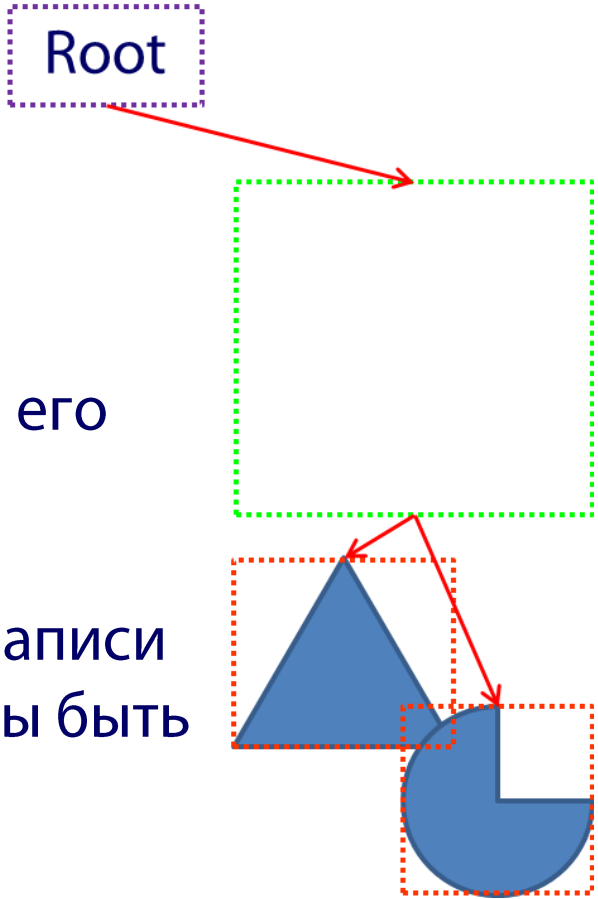


link

R-Tree: удаление

Полный лист

- Если в узле осталось >= m записей, то ничего делать не нужно
- У узла-родителя может понадобится обновить MBR для ребенка, а может и нет
- Q: почему?



Неполный лист

- Если в узле осталось < m записей, удаляем его полностью
- Модифицируем родительский узел
- Заново вставляем по одному оставшиеся записи
- Многие узлы (в которые вставляем) должны быть в кэше/памяти, поэтому быстро доступны
- Q: почему?

Список литературы

1. Antonin Guttman, R-Trees: A Dynamic Index Structure for Spatial Searching, Proc. 1984 ACM SIGMOD International Conference on Management of Data, pp. 47–57.

Некоторые рисунки в данной презентации из [1]

Резюме лекции

План лекции

Часть I. Пространственные данные (spatial data) – **короткий обзор**

Что это? Почему важно? Какие особенности? Где узнать больше?

1. Мотивация: разнообразие типов данных
2. Актуальность & ценность данных (6V)
3. Квалификация для работы с данными: пришел, увидел, победил?
   - пример задачи с R-деревом (пространственная СД)
   - Geoapplications Development (курс Разработка Геоприложений)

Часть II. Структура данных «Классическое R-дерево» (1984 г.)

≈ B-tree для spatial data ⇒ рассматриваем R-tree после B-tree

1. «Постановка задачи»
2. Идеи устройства R-tree
3. Поиск и вставка в R-tree
4. Разбиение узлов R-tree
5. Удаление из R-tree

ДЗ по пространственным СД:
на 8 баллов («отлично»)
не нужно реализовывать R-tree

NATIONAL RESEARCH UNIVERSITY

Благодарю за внимание!

Рамон Антонио Родригес Залепинос
arodriges@hse.ru

Apple нанимает более 70 экспертов в области геоданных

- Apple has posted more than 70 job listings related to mapping technology in the last month.
- Apple needs map tech experts to become a leader in augmented reality.
- The company is ramping up hiring around maps and navigational tech.

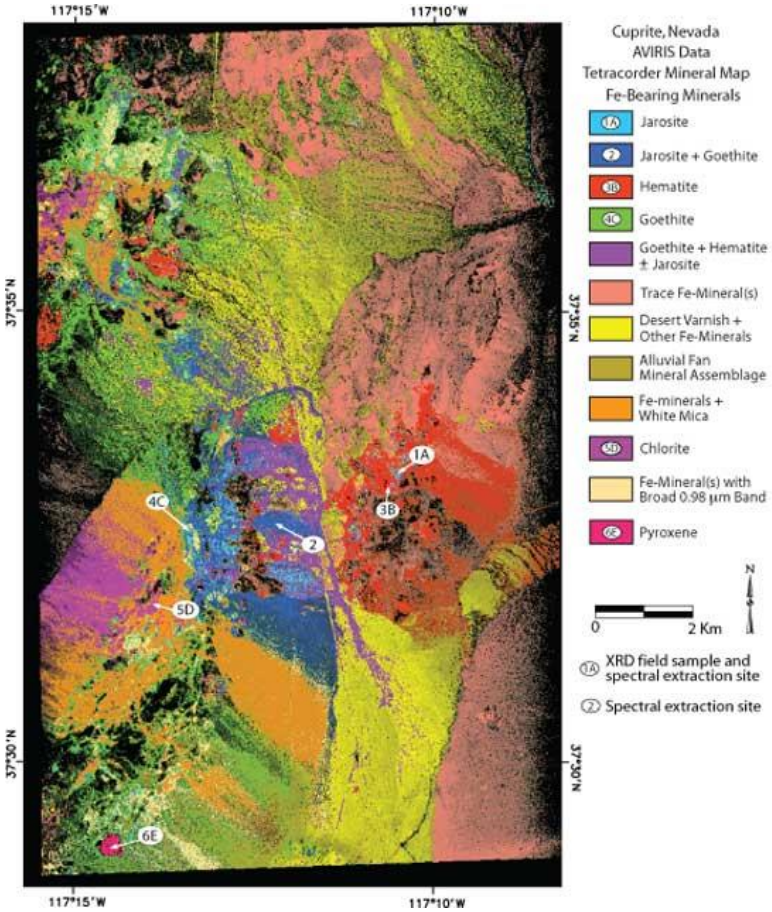
Lora Kolodny | @lorakolodny

Published 5:31 PM ET Wed, 2 Aug 2017 | Updated 7:21 PM ET Wed, 2 Aug 2017

CNBC

https://www.cnbc.com/2017/08/02/apple-has-over-70-map-tech-job-openings.html

IBM ищет минералы по спутниковым данным

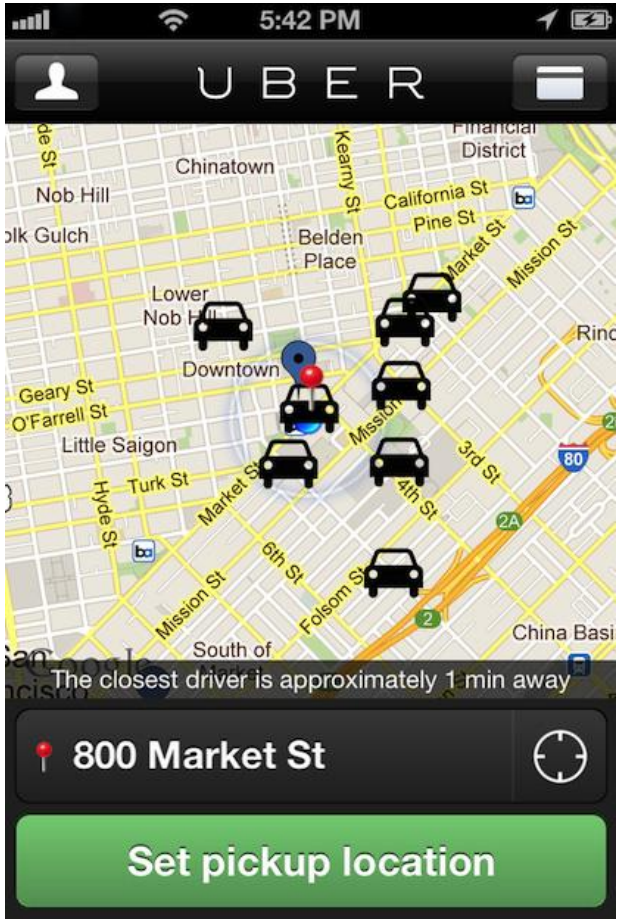


https://www.photonics.com/Articles/Measuring_the_Earth_from_Above_30_years_and/a47298

Uber запустила свою систему визуализации геоданных

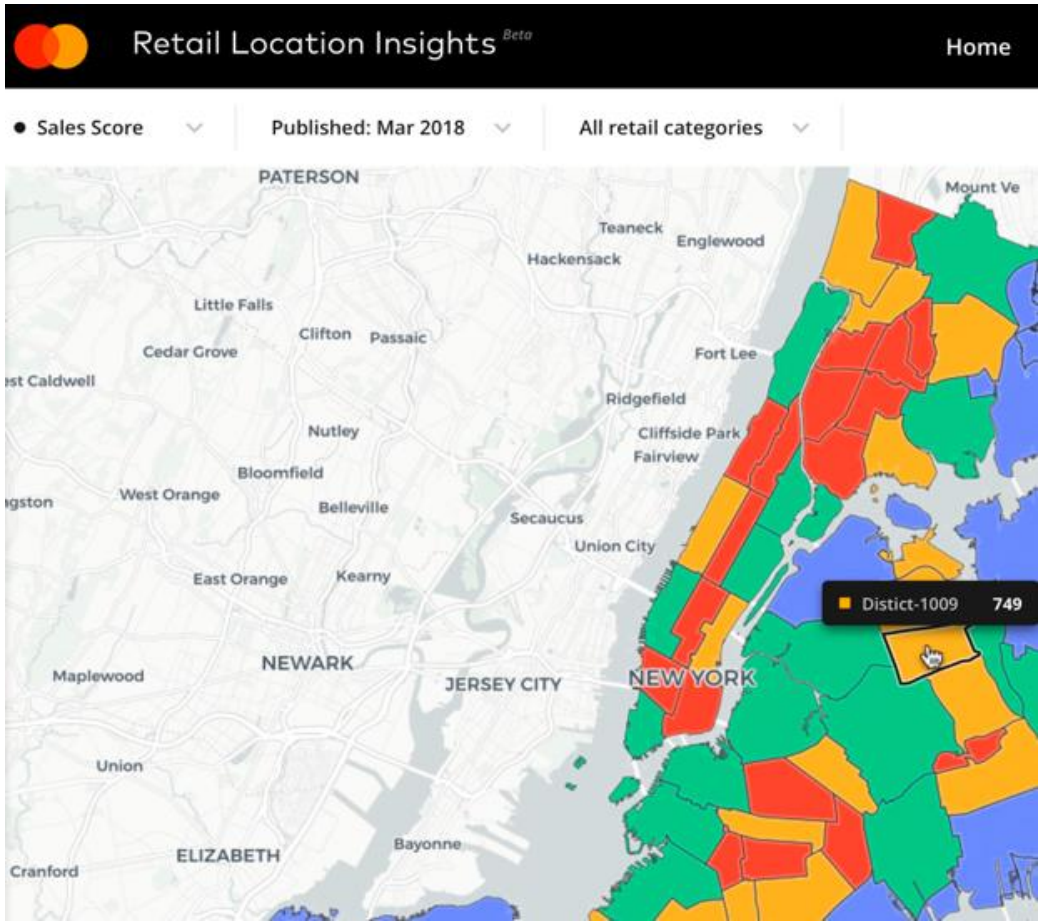
https://eng.uber.com/keplergl/

https://eng.uber.com/engineering

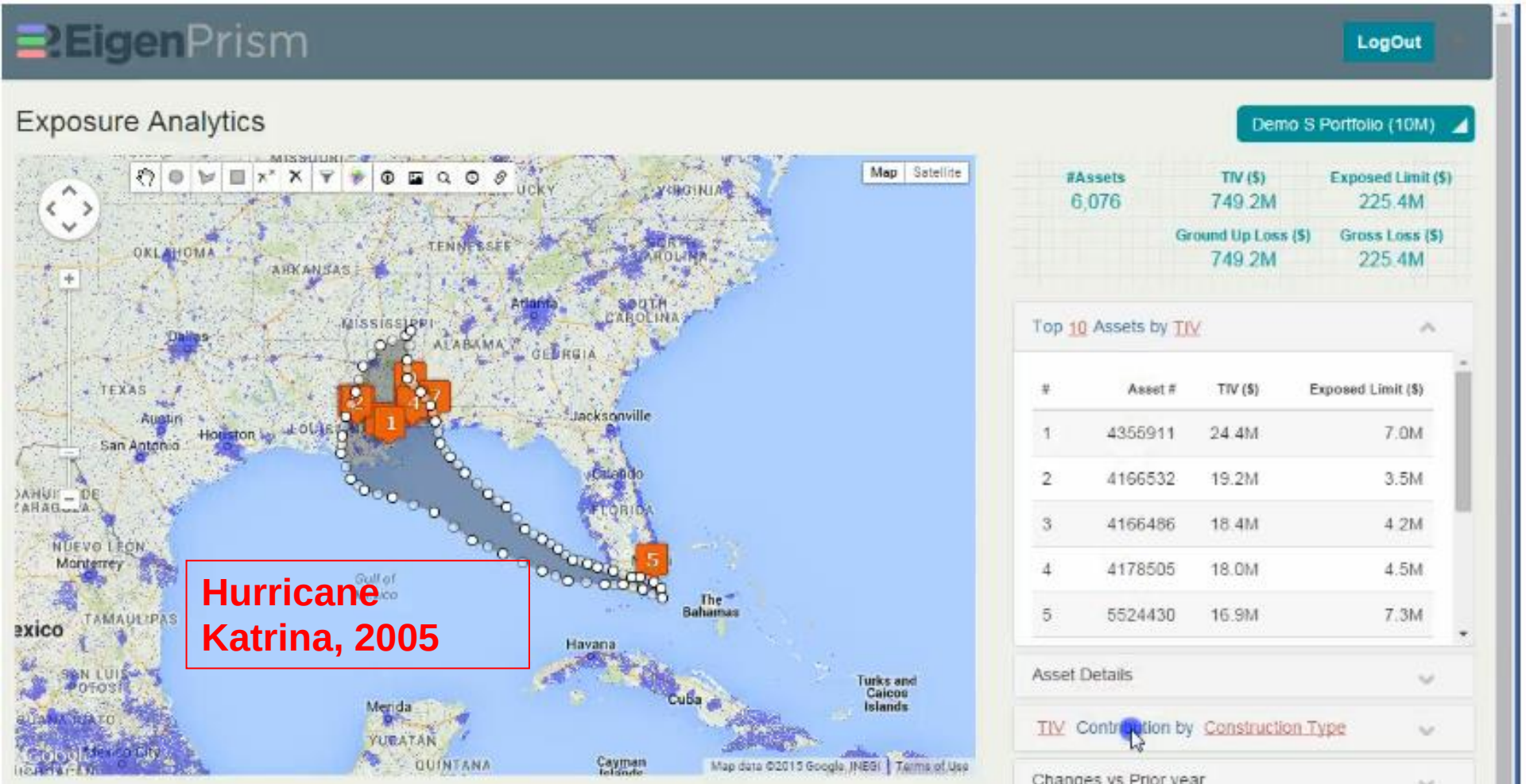


MasterCard: новая система анализа продаж по геоданным

https://carto.com/blog/carto-mastercard-partnering-location-intelligence-solution/



EigenRisk

EigenRisk

Какие дома будут подвержены очередному наводнению и каков прогнозируемый убыток?