



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

G06F 16/904 (2024.01)

(21)(22) Заявка: 2023126040, 11.10.2023

(24) Дата начала отсчета срока действия патента:
11.10.2023

Дата регистрации:
06.02.2024

Приоритет(ы):

(22) Дата подачи заявки: 11.10.2023

(45) Опубликовано: 06.02.2024 Бюл. № 4

Адрес для переписки:

117997, Москва, ул. Вавилова, 19, ПАО
Сбербанк, Правовой департамент

(72) Автор(ы):

Оберемок Андрей Александрович (RU),
Оберемок Надежда Леонидовна (RU),
Козицкий Денис Михайлович (RU),
Димитриади Георгий Гурамович (RU),
Маставичус Владас Прано (RU),
Царев Никита Алексеевич (RU),
Захаров Дмитрий Евгеньевич (RU),
Приходченко Дмитрий Сергеевич (RU),
Аксенова Вера Александровна (RU)

(73) Патентообладатель(и):

Общество с ограниченной ответственностью
"Технологии Отраслевой Трансформации"
(RU)

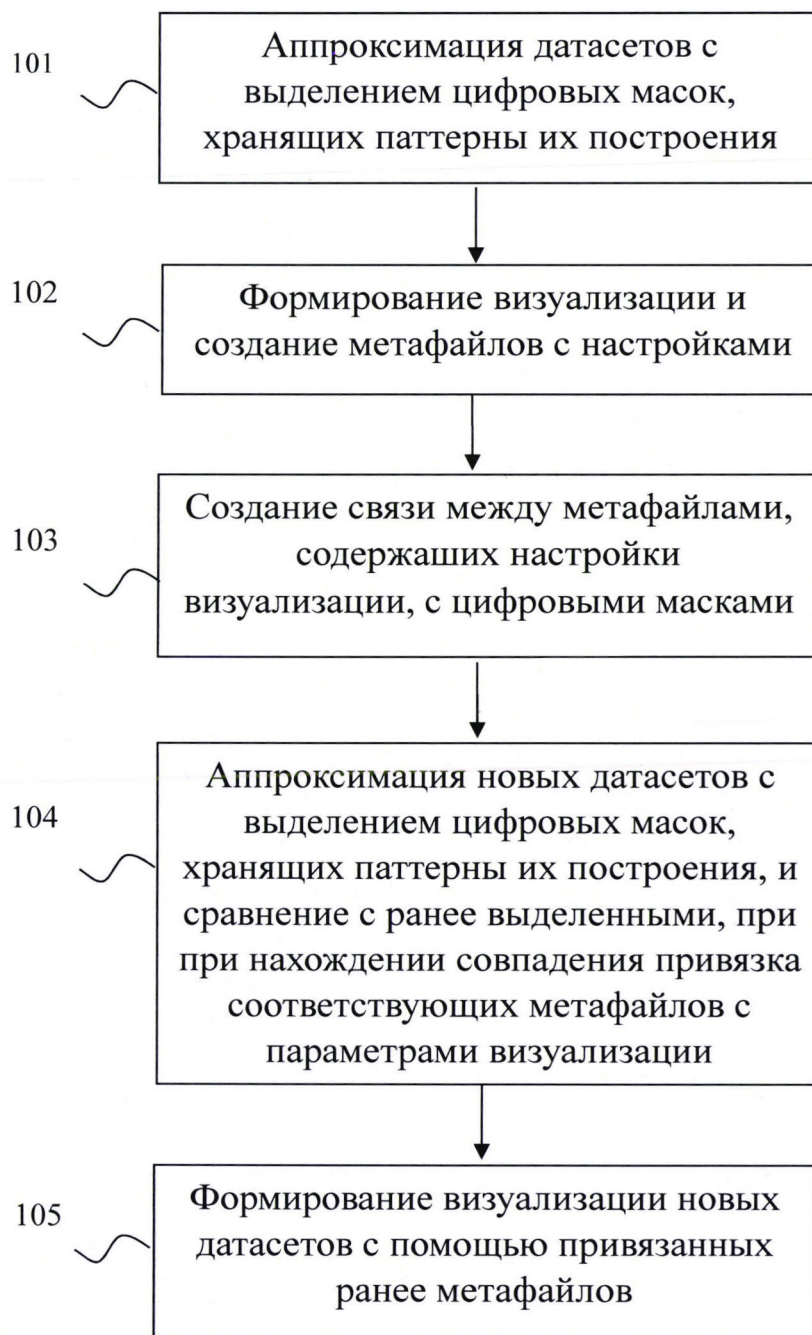
(56) Список документов, цитированных в отчете
о поиске: US 20090016641 A1, 15.01.2009. WO
2017111197 A1, 29.06.2017. US 20180225416 A1,
09.08.2018. RU 2488159 C2, 20.07.2013. RU
2017102380 A, 19.08.2019.

(54) СПОСОБ И СИСТЕМА АВТОМАТИЧЕСКОЙ ВИЗУАЛИЗАЦИИ БОЛЬШИХ МАССИВОВ
ДАННЫХ С ПОМОЩЬЮ ПОИСКА И ВЫДЕЛЕНИЯ ИХ ПАТТЕРНОВ ПОСТРОЕНИЯ

(57) Реферат:

Изобретение относится к способу и системе автоматической визуализации больших массивов данных. Техническим результатом является расширение арсенала технических средств автоматической визуализации больших массивов данных с помощью поиска и выделения их паттернов построения, повышение точности визуализации результатов анализа больших массивов данных. Технический результат достигается благодаря осуществлению способа автоматической визуализации больших массивов данных с помощью поиска и выделения их паттернов построения, в котором: получают датасеты и аппроксимируют их, выделяя упрощенное представление исходных данных и

получая цифровые маски, хранящие паттерны их построения; по представленным датасетам строится визуализация и формируются метафайлы с соответствующими параметрами визуализации; метафайлы связываются с ранее полученными цифровыми масками; при получении нового датасета его аппроксимируют и получают цифровую маску, которую сравнивают с ранее сформированными цифровыми масками и при нахождении совпадения привязывают соответствующие метафайлы с параметрами визуализации; формируют визуализацию нового датасета с помощью привязанных на предыдущем шаге метафайлов. 2 н. и 12 з.п. ф-лы, 9 ил.



ФИГ. 1



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) **ABSTRACT OF INVENTION**

(52) CPC
G06F 16/904 (2024.01)

(21)(22) Application: **2023126040**, 11.10.2023

(24) Effective date for property rights:
11.10.2023

Registration date:
06.02.2024

Priority:

(22) Date of filing: 11.10.2023

(45) Date of publication: 06.02.2024 Bull. № 4

Mail address:
117997, Moskva, ul. Vavilova, 19, PAO Sberbank,
Pravovoj departament

(72) Inventor(s):

Oberemok Andrej Aleksandrovich (RU),
Oberemok Nadezhda Leonidovna (RU),
Kozitskij Denis Mikhajlovich (RU),
Dimitriadi Georgij Guramovich (RU),
Mastavichus Vldas Prano (RU),
Tsarev Nikita Alekseevich (RU),
Zakharov Dmitrij Evgenevich (RU),
Prihodchenko Dmitrij Sergeevich (RU),
Aksenova Vera Aleksandrovna (RU)

(73) Proprietor(s):

Obshchestvo s ogranichennoj otvetstvennostyu
"Tekhnologii Otrasevoj Transformatsii" (RU)

(54) **METHOD AND SYSTEM FOR AUTOMATIC VISUALIZATION OF LARGE DATA ARRAYS USING SEARCH AND ISOLATION OF THEIR CONSTRUCTION PATTERNS**

(57) Abstract:

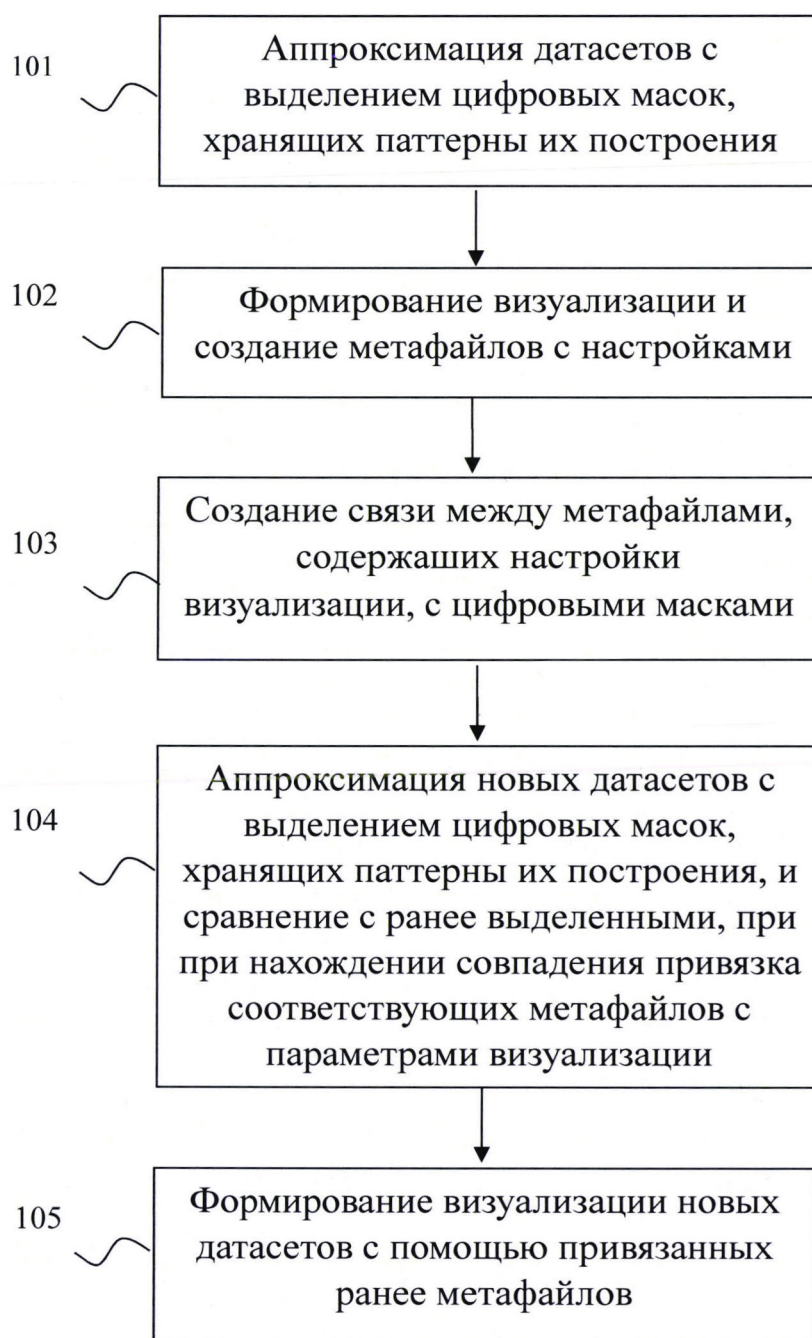
FIELD: data processing.

SUBSTANCE: method and system for automatic visualization of large data arrays. Technical result is achieved by implementing a method for automatically visualizing large data arrays by searching and highlighting their construction patterns, in which: datasets are obtained and approximated, highlighting a simplified representation of the source data and obtaining digital masks that store their construction patterns; based on the presented datasets, a visualization is built and metafiles with the corresponding visualization parameters are generated; metafiles are associated with previously obtained digital masks; when

a new dataset is received, it is approximated and a digital mask is obtained, which is compared with previously generated digital masks and, if a match is found, the corresponding metafiles with visualization parameters are linked; form a visualization of a new dataset using the metafiles linked in the previous step.

EFFECT: expanding the scope of technical means for automatically visualizing large amounts of data by searching and highlighting their construction patterns, increasing the accuracy of visualization of the results of analyzing large data arrays.

14 cl, 9 dwg



ФИГ. 1

ОБЛАСТЬ ТЕХНИКИ

[1] Настоящее техническое решение, в общем, относится к области вычислительной техники, а именно к способу автоматической визуализации больших массивов данных с помощью поиска и выделения их паттернов построения.

УРОВЕНЬ ТЕХНИКИ

[2] В настоящее время информационные системы, предлагающие пользователю возможность визуализации при анализе данных, т.е. функционал создания графиков, описывающих данные, содержат в себе ограниченный набор типов графиков, например: круговая диаграмма, временной график, сводная таблица, воронка и т.д. Далее пользователь самостоятельно выбирает наиболее информативный с его точки зрения тип графика. Данная работа требует высокой квалификации, результат всегда субъективен, и работа может занять продолжительное время.

[3] При этом в существующих подходах к визуализации результатов анализа данных в известных ВІ-системах график, как правило, настраивается для комбинации фильтров, что ограничивает возможности автоматизации процесса визуализации.

[4] Из уровня техники известно техническое решение, описанное в RU 2386172 «ПРАВКА ТЕКСТА ПРОИЗВОЛЬНОЙ ГРАФИКИ ПОСРЕДСТВОМ ИЕРАРХИЧЕСКОГО СПИСКА», 29.06.2005, МАЙКРОСОФТ КОРПОРЕЙШН (US): «способ программного обеспечения графики, который автоматически создает графическое содержание, когда пользователь вводит или модифицирует иерархический список данных содержания и выбирает один из множества вариантов выбора графических определений графических элементов. Кроме того, настоящее изобретение включает в себя пользовательский интерфейс с тремя разными отображениями. Первое отображение представляет пользователю одну или более галерей, включая галерею с множеством графических определений. Второе отображение представляет собой область ввода содержания. И последнее отображение представляет собой полотно для рисования». Описывается инструмент выбора типа и настройки графика человеком, речь об автоматизации не идет.

[5] Из уровня техники известно техническое решение, описанное в RU 2488159 «РАНЖИРОВАНИЕ ТИПОВ ВИЗУАЛИЗАЦИИ НА ОСНОВАНИИ ПРИГОДНОСТИ ДЛЯ ВИЗУАЛИЗАЦИИ НАБОРА ДАННЫХ», 30.03.2009, МАЙКРОСОФТ КОРПОРЕЙШН (US): «Генерируют метаданные визуализации для каждого из множества различных типов визуализации, причем метаданные визуализации содержат данные, описывающие один или более атрибутов типа визуализации, конкретные оси и конкретную последовательность данных для каждого конкретного типа визуализации. Генерируют метаданные набора данных для набора данных, причем метаданные набора данных содержат данные, описывающие один или более атрибутов упомянутого набора данных. Вычисляют оценки пригодности для каждого из множества различных типов визуализации на основании метаданных визуализации и метаданных набора данных, при ранжировании множества различных типов визуализации на основании вычисленных оценок пригодности. Определяют превышает ли каждая из вычисленных оценок пригодности порог. Отображают на интерфейсе пользователя визуальные представления каждого из множества различных типов визуализации, когда вычисленная оценка пригодности превышает порог. Принимают выбор одного из отображенных визуальных представлений и представляют набор данных, используя типы визуализации, соответствующие выбранному визуальному представлению».

При этом в данном решении проводится анализ графиков из ограниченного множества и анализируются их метаданные, что ограничивает возможности

автоматизации.

[6] Из уровня техники известна также библиотека Python AutoViz: выполняет визуализацию в виде диаграмм из плоского текстового файла. На вход библиотеке подается файл (Excel, CSV, txt или json), и AutoViz визуализирует его диаграммами и графиками. Своего интерактивного графического интерфейса не имеет. Взаимодействие с библиотекой происходит программно, через входные параметры.

[7] Недостатками данных решений являются специфические условия для автоматизации процесса визуализации результатов анализа больших массивов данных, либо отсутствие автоматизации, кроме этого существующие решения производят визуализацию недостаточно точно, с существенными искажениями.

РАСКРЫТИЕ ИЗОБРЕТЕНИЯ

[8] Данное техническое решение направлено на устранение недостатков, присущих существующим решениям, известным из уровня техники.

[9] Решаемой технической проблемой в данном техническом решении является обеспечение возможности автоматической визуализации больших массивов данных с помощью поиска и выделения их паттернов построения.

[10] Основным техническим результатом, проявляющимся при решении вышеуказанной проблемы, является расширение арсенала технических средств путем обеспечения возможности автоматической визуализации больших массивов данных с помощью поиска и выделения их паттернов построения. Дополнительным техническим результатом является повышение точности визуализации результатов анализа больших массивов данных.

[11] Указанные технические результаты достигаются благодаря осуществлению способа автоматической визуализации больших массивов данных с помощью поиска и выделения их паттернов построения, реализуемого с помощью процессора и устройства хранения данных, включающего следующие шаги:

- получают датасеты и аппроксимируют их, выделяя упрощенное представление исходных данных и получая цифровые маски, хранящие паттерны их построения;
- по представленным датасетам строится визуализация и формируются метафайлы с соответствующими параметрами визуализации;
- сформированные на предыдущем шаге метафайлы связываются с ранее полученными цифровыми масками;
- при получении нового датасета для построения визуализации его аппроксимируют и получают цифровую маску;
- полученную цифровую маску сравнивают с ранее сформированными цифровыми масками и при нахождении совпадения привязывают соответствующие метафайлы с параметрами визуализации;
- формируют визуализацию нового датасета с помощью привязанных на предыдущем шаге метафайлов.

[12] В одном из частных примеров осуществления способа метафайлы хранят параметры визуализации с исключением пиковых значений.

[13] В другом частном примере осуществления способа метафайлы хранят параметры визуализации с транспонированием.

[14] В другом частном примере осуществления способа метафайлы хранят параметры визуализации с дополнением тренда характеристиками интенсивности изменения.

[15] В другом частном примере осуществления способа метафайлы хранят параметры визуализации с совмещением периодов.

[16] В другом частном примере осуществления способа метафайлы хранят параметры

визуализации с подбором типа шкалы по осям.

[17] В другом частном примере осуществления способа метафайлы хранят параметры визуализации с установкой минимальных и максимальных ограничений по осям.

[18] Кроме того, заявленные технические результаты достигаются за счет системы автоматической визуализации больших массивов данных с помощью поиска и выделения их паттернов построения, содержащей:

- по меньшей мере одно устройство обработки данных;
- по меньшей мере одно устройство хранения данных;
- по меньшей мере одну программу, где одна или более программ хранятся на одном или более устройствах хранения данных и исполняются на одном и более устройствах обработки данных, причем одна или более программ обеспечивает выполнение следующих шагов:

- получают датасеты и аппроксимируют их, выделяя упрощенное представление исходных данных и получая цифровые маски, хранящие паттерны их построения;

- по представленным датасетам строится визуализация и формируются метафайлы с соответствующими параметрами визуализации;

- сформированные на предыдущем шаге метафайлы связываются с ранее полученными цифровыми масками;

- при получении нового датасета для построения визуализации его аппроксимируют и получают цифровую маску;

- полученную цифровую маску сравнивают с ранее сформированными цифровыми масками и при нахождении совпадения привязывают соответствующие метафайлы с параметрами визуализации;

- формируют визуализацию нового датасета с помощью привязанных на предыдущем шаге метафайлов.

[19] В одном из частных примеров осуществления системы метафайлы хранят параметры визуализации с исключением пиковых значений.

[20] В другом частном примере осуществления системы метафайлы хранят параметры визуализации с транспонированием.

[21] В другом частном примере осуществления системы метафайлы хранят параметры визуализации с дополнением тренда характеристиками интенсивности изменения.

[22] В другом частном примере осуществления системы метафайлы хранят параметры визуализации с совмещением периодов.

[23] В другом частном примере осуществления системы метафайлы хранят параметры визуализации с подбором типа шкалы по осям.

[24] В другом частном примере осуществления системы метафайлы хранят параметры визуализации с установкой минимальных и максимальных ограничений по осям.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

[25] Признаки и преимущества настоящего технического решения станут очевидными из приводимого ниже подробного описания и прилагаемых чертежей, на которых:

Фиг. 1 иллюстрирует блок-схему выполнения заявленного способа.

Фиг. 2 иллюстрирует пример упрощенного представления графика.

Фиг. 3 иллюстрирует пример построения цифровой маски на основе датасета.

Фиг. 4 иллюстрирует пример аппроксимации датасета.

Фиг. 5 иллюстрирует пример упрощенной маски датасета.

Фиг. 6 иллюстрирует примеры графика с пиктограммами, показывающими интенсивность изменения (роста или падения) показателя.

Фиг. 7 иллюстрирует месячную историю некоторого показателя в течение 2 лет.

Фиг. 8 иллюстрирует пример совмещенной визуализации.

Фиг. 9 иллюстрирует систему для реализации заявленного способа.

ОСУЩЕСТВЛЕНИЕ ИЗОБРЕТЕНИЯ

[26] Ниже будут описаны термины и понятия, необходимые для реализации настоящего технического решения.

[27] Датасет или набор данных (data set или dataset) - это коллекция данных. В случае с табличными данными набор данных соответствует одной или нескольким таблицам баз данных, где каждый столбец таблицы соответствует отдельной переменной, и каждая строка соответствует записи в наборе данных. Наборы данных хранят значения для каждой переменной, например высоту и вес объекта, для каждого члена набора данных. Наборы данных могут также состоять из коллекции документов или файлов.

[28] Заявленное техническое решение может выполняться, например системой, машиночитаемым носителем, сервером и т.д. В данном техническом решении под системой подразумевается в том числе компьютерная система, ЭВМ (электронно-вычислительная машина), ЧПУ (числовое программное управление), ПЛК (программируемый логический контроллер), компьютеризированные системы управления и любые другие устройства, способные выполнять заданную, четко определенную последовательность операций (действий, инструкций).

[29] Под устройством обработки команд подразумевается электронный блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (программы).

[30] Устройство обработки команд считывает и выполняет машинные инструкции (программы) с одного или более устройств хранения данных, например таких устройств как оперативно запоминающие устройства (ОЗУ) и/или постоянные запоминающие устройства (ПЗУ). В качестве ПЗУ могут выступать, включая, но не ограничиваясь, жесткие диски (HDD), флеш-память, твердотельные накопители (SSD), оптические носители данных (CD, DVD, BD, MD и т.п.) и др.

[31] Программа - последовательность инструкций, предназначенных для исполнения устройством управления вычислительной машины или устройством обработки команд.

[32] Термин «инструкции», используемый в этой заявке, может относиться, в общем, к программным инструкциям или программным командам, которые написаны на заданном языке программирования для осуществления конкретной функции, такой как, например, получение и обработка данных, формирование профиля пользователя, прием и передача сигналов, анализ принятых данных, идентификация пользователя и т.п. Инструкции могут быть осуществлены множеством способов, включающих в себя, например, объектно-ориентированные методы. Например, инструкции могут быть реализованы, посредством языка программирования C++, Java, Python, различных библиотек (например, Microsoft Foundation Classes) и т.д. Инструкции, осуществляющие процессы, описанные в этом решении, могут передаваться как по проводным, так и по беспроводным каналам передачи данных, например, Wi-Fi, Bluetooth, USB, WLAN, LAN и т.п.

[33] Представленный способ автоматической визуализации больших массивов данных с помощью поиска и выделения их паттернов построения (на Фиг. 1 представлена схема способа) решает задачи обеспечения возможности автоматической визуализации больших массивов данных с помощью поиска и выделения их паттернов построения и повышения точности визуализации результатов анализа больших массивов данных за счет последовательного выполнения следующих шагов:

- получают датасеты и аппроксимируют их, выделяя упрощенное представление

исходных данных и получая цифровые маски, хранящие паттерны их построения;

- по представленным датасетам строится визуализация и формируются метафайлы с соответствующими параметрами визуализации;

- сформированные на предыдущем шаге метафайлы связываются с ранее

5 полученными цифровыми масками;

- при получении нового датасета для построения визуализации его аппроксимируют и получают цифровую маску;

- полученную цифровую маску сравнивают с ранее сформированными цифровым масками и при нахождении совпадения привязывают соответствующие метафайлы с

10 параметрами визуализации;

- формируют визуализацию нового датасета с помощью привязанных на предыдущем шаге метафайлов.

[34] На первом этапе способа датасет «упрощается» и это упрощенное представление графика является ключом настройки (в паре «ключ настройки - значение настройки»).

15 Упрощение происходит на основе методик и алгоритмов аппроксимации данных. Выбор конкретного способа аппроксимации исходного датасета не важен для описываемого технического решения. Аппроксимация датасета необходима для получения упрощенного представления исходных данных с тем, чтобы на основе такого представления автоматизировать повторное использование оптимальных настроек

20 графического представления данных. Автоматизация достигается за счет того, что аппроксимированные данные становятся цифровой маской исходных данных и обладают следующими свойствами:

- построены на основе исходных данных;

- отражают характерные особенности исходных данных;

25 • имеют малый размер для удобства хранения и построения связи с настройками визуализации.

[35] Полученная именно таким образом, цифровая маска будет являться ключом настройки. При этом непосредственное применение функций свертки (хеш-функций) к исходным данным, без аппроксимации, даст уникальный отпечаток именно этих данных

30 и не позволит проводить сравнение датасетов, с целью выявить подобие.

[36] В описываемом решении искомый ключ настройки, с одной стороны, является непосредственным образом аппроксимированного датасета, сохраняет сведения о его характерных особенностях, а с другой стороны, не является строгим идентификатором исходного датасета. Нахождение подобия по полученным таким образом

35 идентификаторам позволяет распознавать данные, имеющие схожие по картине распределения значения, наличие периодической цикличности, другие характерные особенности и автоматически применять для визуализации ранее подобранные оптимальные настройки. Например, упрощенному представлению графика на Фиг. 2 соответствует цифровая маска:

40

45

```

    {
    Scale_X: 8,
5    Scale_Y: 8,
    Values: [
        [1, 1],
10        [1, 2],
        [2, 3],
        [3, 4],
15        [3, 5],
        [4, 6],
        [5, 5],
20        [5, 4],
        [6, 3],
        [7, 2],
25        [1, 2]
    ]
    }

```

30 Это значение приблизительно отражает состав исходных данных, при этом оно достаточно компактно, что позволяет использовать его в качестве ключа для построения связи с набором настроек визуализации.

Например, на Фиг. 2, шкалы X и Y разбиваются на N частей и строится «пиксельное» представление графика - если график попадает в квадрат, то значение квадрата становится равным 1. Таким образом, в этом примере датасет «упрощается» до 35 двумерного массива N*N и от этого упрощенного представления легко и быстро посчитать хэш-значение. В примере на Фиг. 2 проиллюстрировано загруженное (пикселизированное) представление графика функции синус. Или, например, от графика берется производная, значение которой может быть, всегда положительно. Или, например, от графика можно взять минимальные и максимальные значения шкал.

40 Изначально, когда для анализируемого датасета нет (или не найдено) метаданных настройки, визуализация происходит по исходным данным «как есть» без применения каких-либо настроек. Далее, ознакомившись с таким первоначальным графиком, пользователь может воспользоваться системой поиска более оптимального (удобного для анализа) отображения графика данных, включая:

- 45
- исключение пиковых значений;
 - транспонирование;
 - дополнение тренда характеристиками интенсивности изменения;
 - совмещение периодов;

- подбор типа шкалы по осям;
- установка min/max-ограничений по осям.

Полученные на данном этапе настройки визуализации сохраняются, ключом поиска для их повторного использования становится цифровая маска исходного датасета:

[цифровая маска] => [набор настроек]

Например: см. таблицу ниже (связь Маска ->Настройка)

Рассмотрим пример построения цифровой маски на основе датасета, представленного на Фиг. 3.

Аппроксимируем данный датасет с шагом 4 по осям X и Y (на Фиг. 4).

Таким образом, мы получим упрощенную маску исходного датасета, удовлетворяющую нашим требованиям - близкую к исходному, но более простую: представлено на Фиг. 5.

Если размер исходного датасета достаточно велик, так, что за одну итерацию аппроксимации с выбранным шагом получается более 1000 точек, операцию следует повторять до получения маски менее 1000 точек.

Для рассматриваемого датасета, связь Маска —> Настройка будет иметь вид:

Цифровая маска	Настройка
<pre> { scale_X: 4, scale_Y: 4, values: [[1, 1], [1, 2], [2, 3], [3, 4], [3, 5], [4, 6], [5, 5], [5, 4], [6, 3], [7, 2], [1, 2]] }</pre>	<pre> { exclude_peaks_above: 5000000, // исключить пиковые значения более 5 млн transpose: false, // не транспонировать intensity_icons: false, // не добавлять значки интенсивности изменения combine_periods: "w2w", // совместить тренды неделя к неделе, возможны: y2y, q2q, m2m, d2d type_scale_Y: "linear", // линейный тип шкалы по оси-Y, возможны: logarithmic, power scale_Y: [0, 5000000] // установить min/max по оси-Y }</pre>

[37] Получение цифровых масок датасетов происходит на основе известных методик и алгоритмов аппроксимации данных, возможно использование готовых библиотек, реализующих необходимый функционал аппроксимации. Функционал поиска и связи реализован на уровне баз данных в языке SQL.

[38] В качестве дополнительного примера практического применения приведем исключение из анализа точек с пиковыми значениями. В ручном режиме исключение точек реализовано через интерактивное взаимодействие пользователя с точками/узлами графика. В автоматическом режиме, с использованием описанного технического решения выявляются точки со значительно отличающимися значениями от основного массива значений. Затем устанавливается порядок размерности отличия: в 10 раз, в 100 раз, и т.д. Если такие точки/узлы находятся, они исключаются (либо предлагаются к исключению) из визуализации графика.

[39] Для выявления точек с пиковыми значениями сначала рассчитывается среднее значение исследуемого показателя в представленных для анализа календарных границах. Затем происходит оценка порядка отклонения значений от среднего по набору данных в целом. Оценка происходит последовательно от большего порядка к меньшему. Если

найжены значения больше/меньше среднего в 1000 раз - они исключаются (или предлагаются к исключению) при визуализации. Если такого порядка отклонений нет, происходит поиск значения с меньшим порядком в 100 раз/10 раз и т.д. При этом у пользователя есть возможность управлять шагом порядка отклонения или остановить

работу алгоритма на произвольном шаге.
[40] В рамках данной задачи, для оценки отклонений допустимо применение готовых библиотек, реализующих распространенные статистические методики. Новизна применения достигается за счет применения данного инструмента уже во время визуализации с возможностью ручной настройки процесса пользователем для поиска наиболее подходящего способа графического представления данных.

[41] В качестве еще одного дополнительного примера практического применения приведем выявление и обозначение на тренде графика характеристик интенсивности изменения.

Выявляются участки графика, имеющие:

- неизменное значение производной (участок с равномерным изменением показателя);
- монотонное изменение производной первого порядка и производная второго порядка неизменна (участок монотонного ускорения/замедления изменения показателя);
- монотонное изменение производной второго порядка и производная третьего порядка неизменна (участок интенсивного ускорения/замедления изменения показателя).

Затем, в соответствии с выявленными участками на графике располагаются пиктограммы, иллюстрирующие интенсивность изменения (роста или падения) показателя (на Фиг. 6).

[42] В качестве дополнительного примера практического применения приведем совмещение сегментов периодических (календарных) трендов для выявления цикличности изменений (либо отклонений цикличности). Например, на Фиг. 7 последовательно показана помесечная история некоторого показателя в течении 2 лет.

Используя описанное техническое решение автоматически выявляется наличие полной ежемесячной истории за 2-летний период и предлагается вариант совмещенной визуализации, см. на Фиг. 8.

Совмещенная визуализация, таким образом, позволит наглядно оценить динамику показателя в сравнении период-к-периоду.

[43] В общем виде (см. Фиг. 9) система автоматической визуализации больших массивов данных с помощью поиска и выделения их паттернов построения (900) содержит объединенные общей шиной информационного обмена один или несколько процессоров (901), средства памяти, такие как ОЗУ (902) и ПЗУ (903) и интерфейсы ввода/вывода (904).

[44] Процессор (901) (или несколько процессоров, многоядерный процессор и т.п.) может выбираться из ассортимента устройств, широко применяемых в настоящее время, например, таких производителей, как: Intel™, AMD™, Apple™, Samsung Exynos™, MediaTek™, Qualcomm Snapdragon™ и т.п. Под процессором или одним из используемых процессоров в системе (900) также необходимо учитывать графический процессор, например, GPU NVIDIA с программной моделью, совместимой с CUDA, или Graphcore, тип которых также является пригодным для полного или частичного выполнения способа, а также может применяться для обучения и применения моделей машинного обучения в различных информационных системах.

[45] ОЗУ (902) представляет собой оперативную память и предназначено для хранения исполняемых процессором (901) машиночитаемых инструкций для выполнения необходимых операций по логической обработке данных. ОЗУ (902), как правило,

содержит исполняемые инструкции операционной системы и соответствующих программных компонент (приложения, программные модули и т.п.). При этом, в качестве ОЗУ (902) может выступать доступный объем памяти графической карты или графического процессора.

5 [46] ПЗУ (903) представляет собой одно или более устройств постоянного хранения данных, например, жесткий диск (HDD), твердотельный накопитель данных (SSD), флэш-память (EEPROM, NAND и т.п.), оптические носители информации (CD-R/RW, DVD-R/RW, BlueRay Disc, MD) и др.

10 [47] Для организации работы компонентов устройства (900) и организации работы внешних подключаемых устройств применяются различные виды интерфейсов В/В (904). Выбор соответствующих интерфейсов зависит от конкретного исполнения вычислительного устройства, которые могут представлять собой, не ограничиваясь: PCI, AGP, PS/2, IrDa, FireWire, LPT, COM, SATA, IDE, Lightning, USB (2.0, 3.0, 3.1, micro, mini, type C), TRS/Audio jack (2.5, 3.5, 6.35), HDMI, DVI, VGA, Display Port, RJ45, RS232 и
15 т.п.

[48] Для обеспечения взаимодействия пользователя с устройством (900) применяются различные средства (905) В/В информации, например, клавиатура, дисплей (монитор), сенсорный дисплей, тач-пад, джойстик, манипулятор мышь, световое перо, стилус, сенсорная панель, трекбол, динамики, микрофон, средства дополненной реальности,
20 оптические сенсоры, планшет, световые индикаторы, проектор, камера, средства биометрической идентификации (сканер сетчатки глаза, сканер отпечатков пальцев, модуль распознавания голоса) и т.п.

[49] Средство сетевого взаимодействия (906) обеспечивает передачу данных посредством внутренней или внешней вычислительной сети, например, Интранет,
25 Интернет, ЛВС и т.п. В качестве одного или более средств (906) может использоваться, но не ограничиваясь: Ethernet карта, GSM модем, GPRS модем, LTE модем, 5G модем, модуль спутниковой связи, NFC модуль, Bluetooth и/или BLE модуль, Wi-Fi модуль и др.

[50] Конкретный выбор элементов устройства (900) для реализации различных
30 программно-аппаратных архитектурных решений может варьироваться с сохранением обеспечиваемого требуемого функционала. В частности, подобная реализация может быть выполнена с помощью электронных компонент, используемых для создания цифровых интегральных схем. Не ограничиваясь, могут быть использоваться микросхемы, логика работы которых определяется при изготовлении, или
35 программируемые логические интегральные схемы (ПЛИС), логика работы которых задается посредством программирования. Для программирования используются программаторы и отладочные среды, позволяющие задать желаемую структуру цифрового устройства в виде принципиальной электрической схемы или программы на специальных языках описания аппаратуры: Verilog, VHDL, AHDL и др. Альтернативой
40 ПЛИС являются: программируемые логические контроллеры (ПЛК), базовые матричные кристаллы (БМК), требующие заводского производственного процесса для программирования; ASIC - специализированные заказные большие интегральные схемы (БИС), которые при мелкосерийном и единичном производстве существенно дороже. Таким образом, реализация может быть достигнута стандартными средствами,
45 базирующимися на классических принципах реализации основ вычислительной техники.

[51] Представленные материалы заявки раскрывают предпочтительные примеры реализации технического решения и не должны трактоваться как ограничивающие иные, частные примеры его воплощения, не выходящие за пределы испрашиваемой

правовой охраны, которые являются очевидными для специалистов соответствующей области техники.

(57) Формула изобретения

- 5 1. Способ автоматической визуализации больших массивов данных с помощью поиска и выделения их паттернов построения, реализуемый с помощью процессора и устройства хранения данных, включающий следующие шаги:
получают датасеты и аппроксимируют их, выделяя упрощенное представление исходных данных и получая цифровые маски, хранящие паттерны их построения;
10 по представленным датасетам строится визуализация и формируются метафайлы с соответствующими параметрами визуализации;
сформированные на предыдущем шаге метафайлы связываются с ранее полученными цифровыми масками;
при получении нового датасета для построения визуализации его аппроксимируют
15 и получают цифровую маску;
полученную цифровую маску сравнивают с ранее сформированными цифровыми масками и при нахождении совпадения привязывают соответствующие метафайлы с параметрами визуализации;
формируют визуализацию нового датасета с помощью привязанных на предыдущем
20 шаге метафайлов.
2. Способ по п. 1, характеризующийся тем, что метафайлы хранят параметры визуализации с исключением пиковых значений.
3. Способ по п. 1, характеризующийся тем, что метафайлы хранят параметры визуализации с транспонированием.
- 25 4. Способ по п. 1, характеризующийся тем, что метафайлы хранят параметры визуализации с дополнением тренда характеристиками интенсивности изменения.
5. Способ по п. 1, характеризующийся тем, что метафайлы хранят параметры визуализации с совмещением периодов.
6. Способ по п. 1, характеризующийся тем, что метафайлы хранят параметры
30 визуализации с подбором типа шкалы по осям.
7. Способ по п. 1, характеризующийся тем, что метафайлы хранят параметры визуализации с установкой минимальных и максимальных ограничений по осям.
8. Система автоматической визуализации больших массивов данных с помощью поиска и выделения их паттернов построения, содержащая:
35 по меньшей мере одно устройство обработки данных;
по меньшей мере одно устройство хранения данных;
по меньшей мере одну программу, где одна или более программ хранятся на одном или более устройствах хранения данных и исполняются на одном и более устройствах обработки данных, причем одна или более программ обеспечивает выполнение
40 следующих шагов:
получают датасеты и аппроксимируют их, выделяя упрощенное представление исходных данных и получая цифровые маски, хранящие паттерны их построения;
по представленным датасетам строится визуализация и формируются метафайлы с
соответствующими параметрами визуализации;
45 сформированные на предыдущем шаге метафайлы связываются с ранее полученными цифровыми масками;
при получении нового датасета для построения визуализации его аппроксимируют и получают цифровую маску;

полученную цифровую маску сравнивают с ранее сформированными цифровыми масками и при нахождении совпадения привязывают соответствующие метафайлы с параметрами визуализации;

5 формируют визуализацию нового датасета с помощью привязанных на предыдущем шаге метафайлов.

9. Система по п. 1, характеризующаяся тем, что метафайлы хранят параметры визуализации с исключением пиковых значений.

10. Система по п. 1, характеризующаяся тем, что метафайлы хранят параметры визуализации с транспонированием.

10 11. Система по п. 1, характеризующаяся тем, что метафайлы хранят параметры визуализации с дополнением тренда характеристиками интенсивности изменения.

12. Система по п. 1, характеризующаяся тем, что метафайлы хранят параметры визуализации с совмещением периодов.

15 13. Система по п. 1, характеризующаяся тем, что метафайлы хранят параметры визуализации с подбором типа шкалы по осям.

14. Система по п. 1, характеризующаяся тем, что метафайлы хранят параметры визуализации с установкой минимальных и максимальных ограничений по осям.

20

25

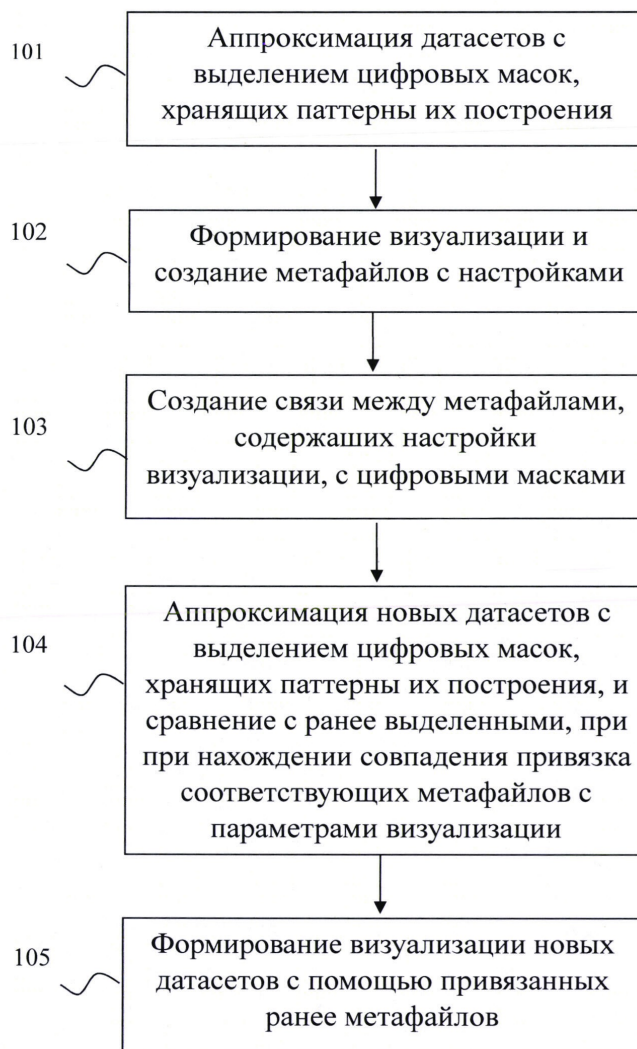
30

35

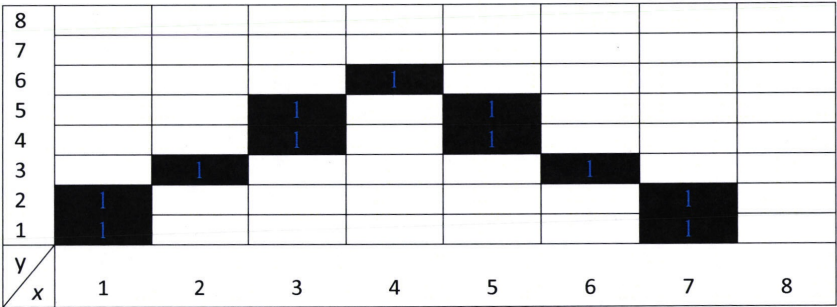
40

45

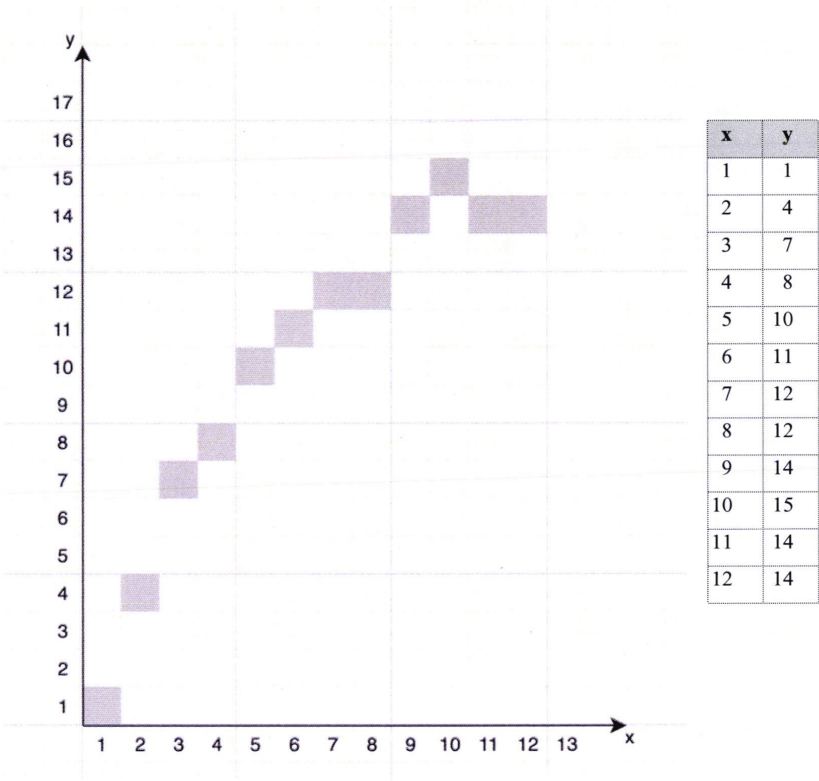
100



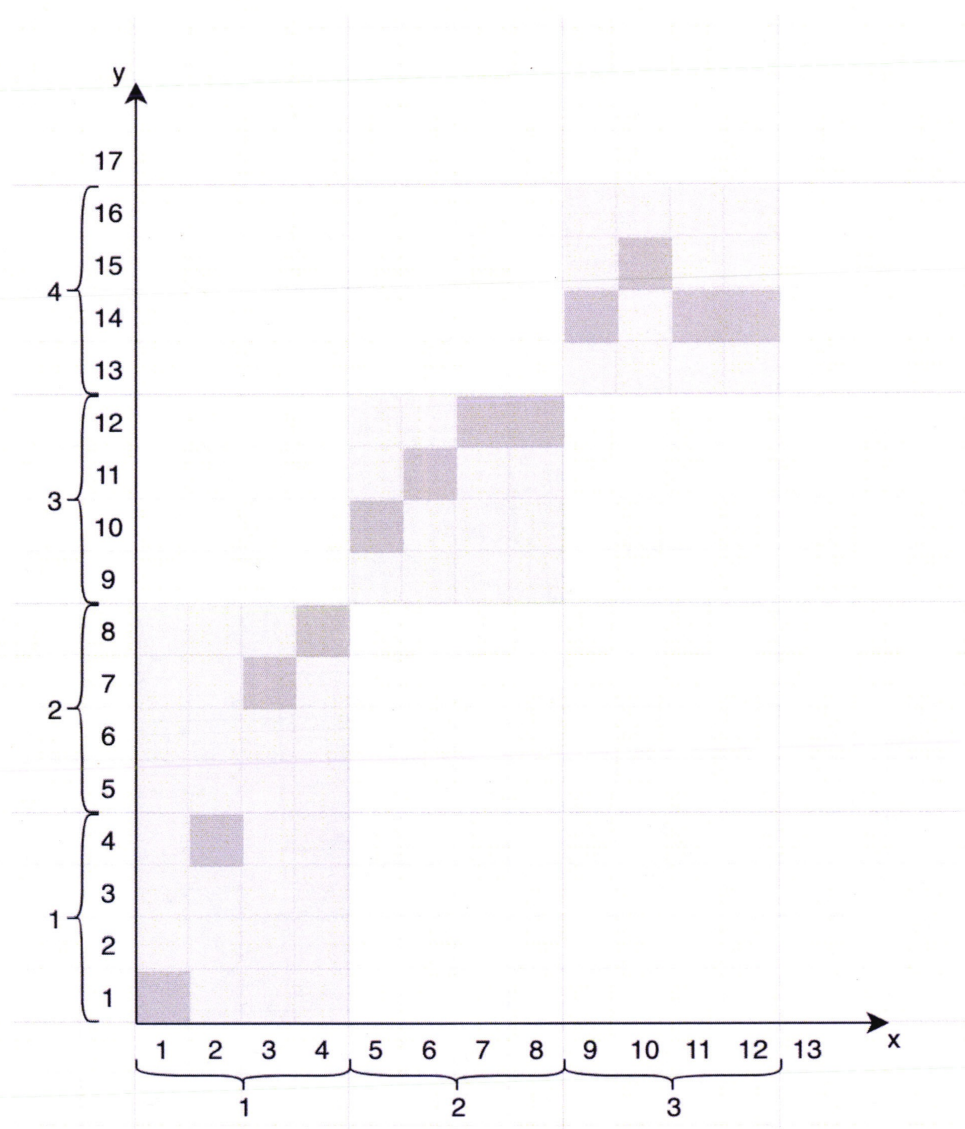
ФИГ. 1



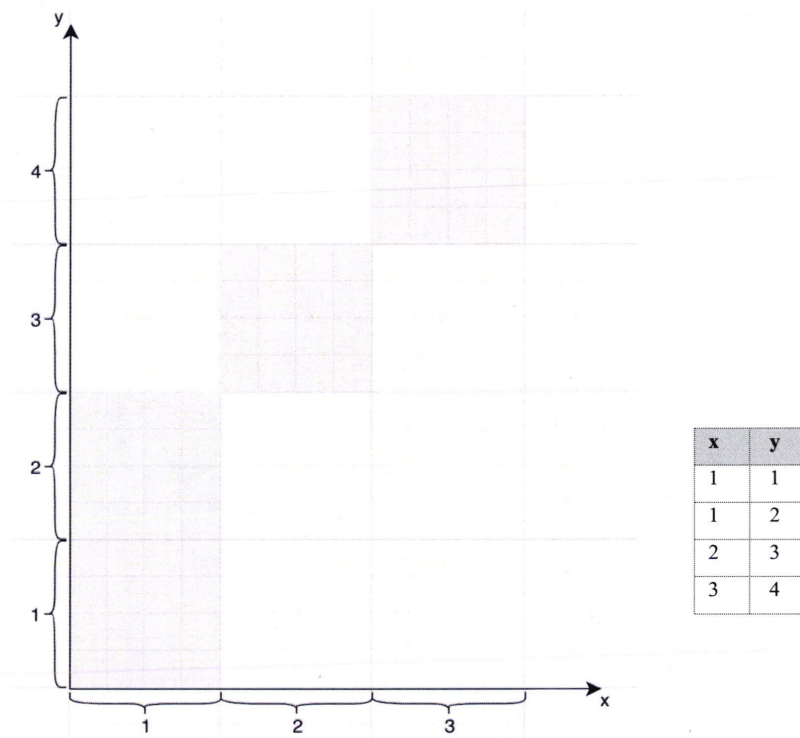
ФИГ. 2



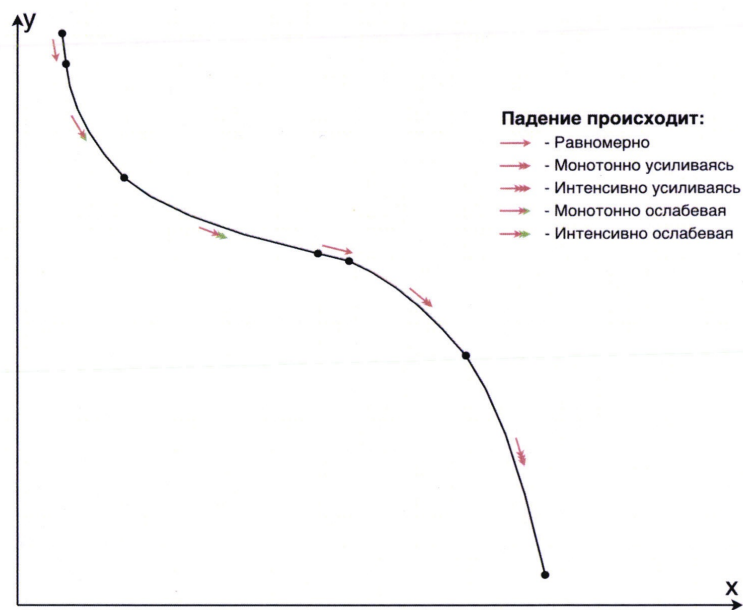
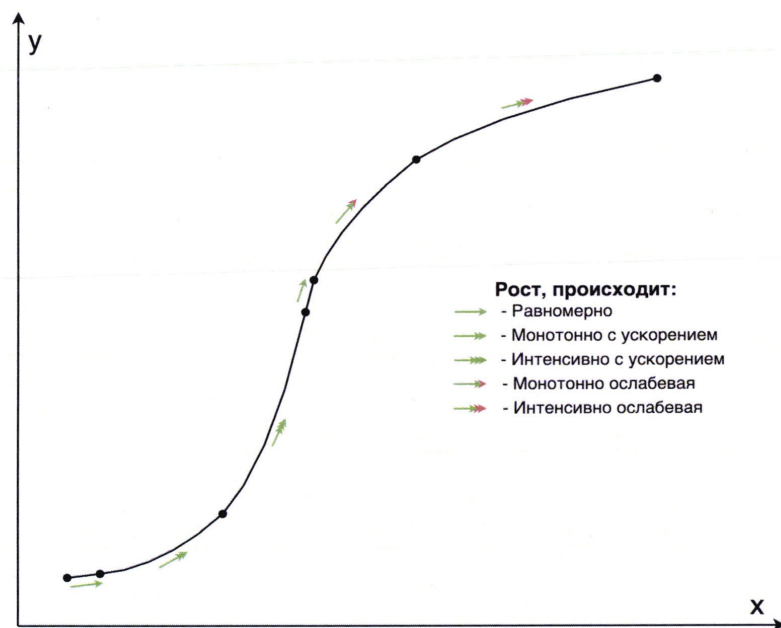
ФИГ. 3



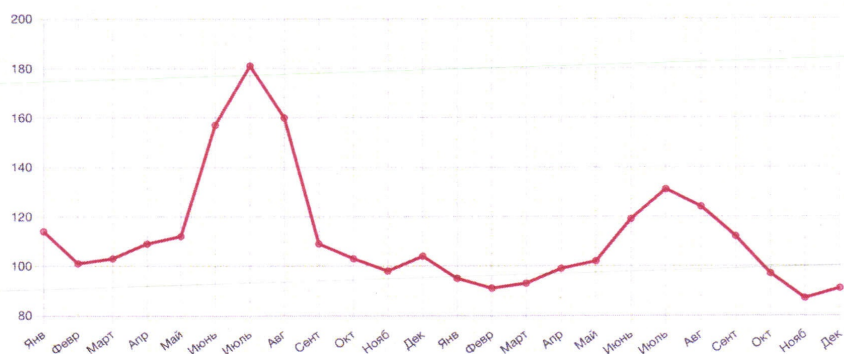
ФИГ. 4



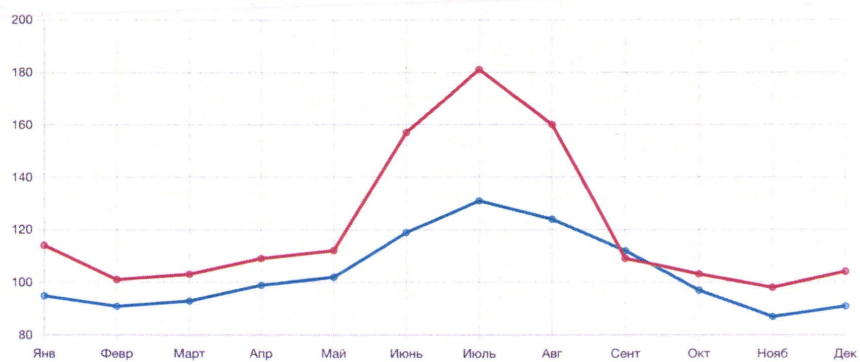
ФИГ. 5



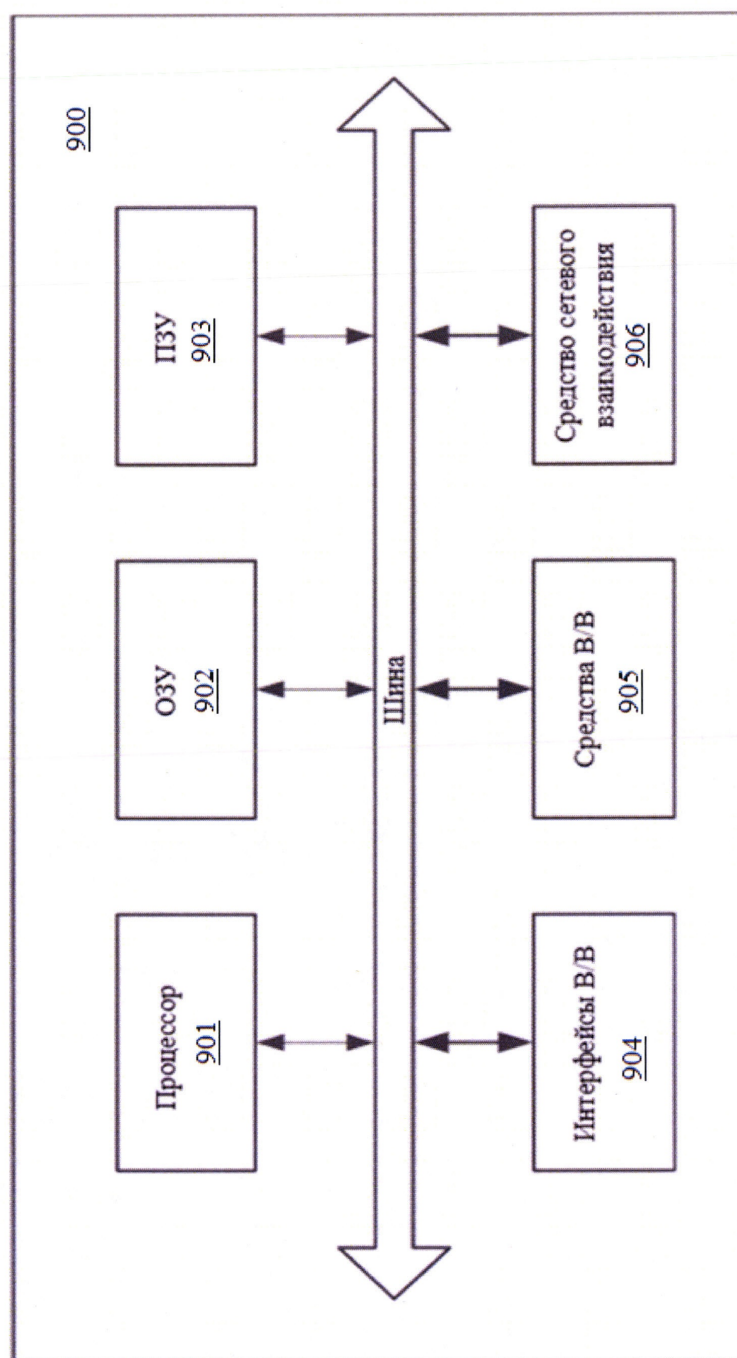
ФИГ. 6



ФИГ. 7



ФИГ. 8



ФИГ. 9