



**ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ**

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2011143804/08, 23.04.2010

(24) Дата начала отсчета срока действия патента:
23.04.2010

Приоритет(ы):

(30) Конвенционный приоритет:
30.04.2009 US 12/433,288

(43) Дата публикации заявки: 10.05.2013 Бюл. № 13

(45) Опубликовано: 20.02.2015 Бюл. № 5

(56) Список документов, цитированных в отчете о поиске: US 2002/0199156 A1, 26.12.2002. US 6529900 B1, 04.03.2003. US 2008/0180458 A1, 31.07.2008. US 5581677 A, 03.12.1996. US 5461708 A, 24.10.1995. RU 2321892 C2, 10.04.2008

(85) Дата начала рассмотрения заявки РСТ на национальной фазе: 28.10.2011

(86) Заявка РСТ:
US 2010/032307 (23.04.2010)

(87) Публикация заявки РСТ:
WO 2010/126802 (04.11.2010)

Адрес для переписки:

129090, Москва, ул. Б. Спасская, 25, строение 3,
ООО "Юридическая фирма Городиский и
Партнеры"

(72) Автор(ы):

АЛЛИН Бэрри Кристофер (US),
РАБЛ Б. Скотт (US)

(73) Патентообладатель(и):

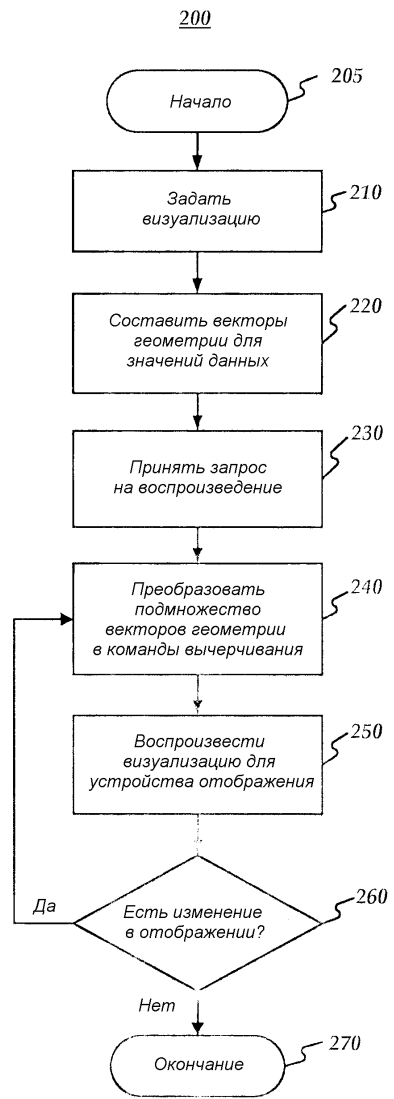
МАЙКРОСОФТ КОРПОРЕЙШН (US)

(54) ОПТИМИЗАЦИЯ ПРОИЗВОДИТЕЛЬНОСТИ ПЛАТФОРМЫ ВИЗУАЛИЗАЦИИ ДАННЫХ

(57) Реферат:

Изобретение относится к компьютерной технике, а именно к способам формирования визуальных объектов. Техническим результатом является повышение производительности выполнения визуализации данных за счет осуществления алгоритма пропуска геометрии в соответствии с предварительно определенными критериями. Предложен способ выполнения визуализации посредством платформы визуализации данных. Способ включает в себя этапы, на которых осуществляют задание визуализации, создание множества векторов

геометрии. А также согласно способу осуществляют прием запроса на отображение визуализации и осуществляют выполнение итерации по множеству векторов геометрии. Далее осуществляют пропуск, по меньшей мере, одного первого вектора из множества геометрических записей согласно алгоритму пропуска геометрии, содержащему, по меньшей мере, один из следующих критериев: обнаружение линии нулевой длины, общее число множества векторов геометрии, общее число точек данных, размер отображения, тип данных и тип



Фиг.2

RU 2541876 C2

RU 2541876 C2



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.
G06F 3/14 (2006.01)
G06F 3/0481 (2013.01)

(12) **ABSTRACT OF INVENTION**

(21)(22) Application: **2011143804/08, 23.04.2010**
 (24) Effective date for property rights:
23.04.2010
 Priority:
 (30) Convention priority:
30.04.2009 US 12/433,288
 (43) Application published: **10.05.2013 Bull. № 13**
 (45) Date of publication: **20.02.2015 Bull. № 5**
 (85) Commencement of national phase: **28.10.2011**
 (86) PCT application:
US 2010/032307 (23.04.2010)
 (87) PCT publication:
WO 2010/126802 (04.11.2010)
 Mail address:
**129090, Moskva, ul. B. Spasskaja, 25, stroenie 3,
OOO "Juridicheskaja firma Gorodisskij i Partnery"**

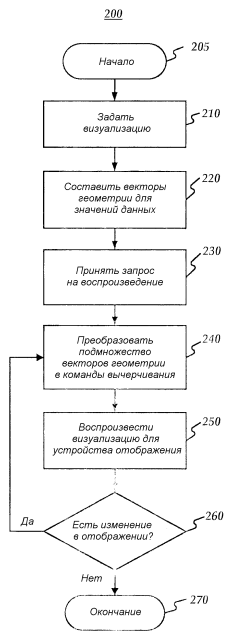
(72) Inventor(s):
**ALLIN Behrri Kristofer (US),
RABL B. Skott (US)**
 (73) Proprietor(s):
MAJKROSOFT KORPOREJShN (US)

(54) **DATA VISUALISATION PLATFORM PERFORMANCE OPTIMISATION**

(57) Abstract:
 FIELD: physics, computer engineering. 13 cl, 3 dwg
 SUBSTANCE: invention relates to computer engineering and specifically to methods of forming visual objects. Disclosed is a method of performing visualisation through a data visualisation platform. The method includes steps of carrying out a visualisation task and creating a plurality of geometry vectors. The method also includes receiving a request to display the visualisation and performing iteration on the plurality of geometry vectors. Further, the method includes skipping at least one of a first vector from the plurality of geometric records according to a geometry skipping algorithm comprising at least one of the following criteria: detecting a zero-length line, the total number of the plurality of geometry vectors, the total number of data points, the display size, data type and visualisation type.
 EFFECT: high performance of visualising data owing to execution of a geometry skipping algorithm in accordance with predefined criteria.

C 2
6
9
7
8
1
4
5
2
R U

R U
2
5
4
1
8
7
6
C 2



Фиг.2

RU 2541876 C2

RU 2541876 C2

ПРЕДШЕСТВУЮЩИЙ УРОВЕНЬ ТЕХНИКИ

Оптимизация производительности платформы визуализации данных обеспечивает улучшенную производительность в формировании визуальных объектов. В некоторых ситуациях, формирование визуальных объектов может являться для компьютера очень интенсивным по производительности. Например, формирование диаграмм и карт может требовать большой величины вычислительной мощности и/или памяти, особенно если объект содержит большое число точек данных. Обычная стратегия состоит в том, чтобы визуально воспроизводить каждую точку данных, независимо от размера результирующего объекта. Это может вызывать проблемы, поскольку некоторые операции над объектом могут вызывать, что компьютер становится медленным для реагирования. Например, отображение большой карты, выборка порции подробной диаграммы, прокрутка, операция печати и/или модификация объекта могут требовать большой величины вычислительной мощности и могут вызывать, что компьютер отвечает медленно.

КРАТКОЕ ОПИСАНИЕ СУЩНОСТИ ИЗОБРЕТЕНИЯ

Может обеспечиваться оптимизация производительности платформы визуализации данных. Это краткое описание приведено, чтобы в упрощенной форме представить подборку понятий, которые дополнительно описаны ниже в подробном описании сущности изобретения. Это краткое описание не предназначено для идентификации ключевых или существенных признаков заявленного объекта изобретения. Также это краткое описание не предназначено, чтобы использоваться для ограничения объема заявленного объекта изобретения.

Может обеспечиваться оптимизация платформы визуализации данных. Приложения могут поставлять значения данных и запрашивать от платформы визуализации данных (DVP) создание визуализации. DVP может составлять множество записей геометрии, связанных с подмножеством значений данных визуализации. Приложение может воспроизводить визуализацию путем выполнения итерации по векторам геометрии и преобразования подмножества векторов в команды вычерчивания (рисования) для вывода на устройство отображения.

И предшествующее общее описание, и последующее подробное описание обеспечивают примеры и являются лишь пояснительными. Соответственно, предшествующее общее описание и последующее подробное описание не следует рассматривать ограничительными. Кроме того, могут обеспечиваться признаки или изменения в дополнение к таковым, изложенным в документе. Например, варианты осуществления могут быть ориентированы на различные комбинации и подкомбинации признаков, описанные в подробном описании.

КРАТКОЕ ОПИСАНИЕ ФИГУР ЧЕРТЕЖЕЙ

Сопроводительные чертежи, которые включены в данное раскрытие и составляют его часть, иллюстрируют различные варианты осуществления настоящего изобретения.

На чертежах:

Фиг.1 - блок-схема операционной среды;

Фиг.2 - структурная схема способа для оптимизации производительности платформы визуализации; и

Фиг.3 - блок-схема системы, включающей в себя вычислительное устройство.

ПОДРОБНОЕ ОПИСАНИЕ СУЩНОСТИ ИЗОБРЕТЕНИЯ

Нижеследующее подробное описание ссылается на сопроводительные чертежи.

Везде, где возможно, одинаковые ссылочные позиции используются на чертежах и нижеследующем описании для ссылок на одинаковые или подобные элементы. Тогда

как могут описываться варианты осуществления изобретения, являются возможными модификации, адаптация и другие реализации. Например, могут выполняться замены, дополнения или модификации по отношению к элементам, проиллюстрированным на чертежах, и способы, описанные в документе, могут быть модифицированы путем замены, изменения порядка следования или добавления этапов к раскрытым способам. Соответственно, нижеследующее подробное описание не ограничивает изобретение. Взамен надлежащий объем изобретения определяется прилагаемой формулой изобретения.

10 Может обеспечиваться оптимизация производительности платформы визуализации данных (DVP). В соответствии с вариантами осуществления настоящего изобретения DVP может пропускать точки данных (точки на графике) в течение составления объекта в соответствии с разрешающей способностью устройства отображения и может создавать в машинной памяти минимальный набор векторов геометрии. DVP может также пропускать составленные векторы в течение воспроизведения (рендеринга) 15 объекта. DVP может дополнительно позволять масштабирование алгоритмов для пропуска точек данных и/или векторов, чтобы предоставлять различные установочные параметры качества, например, путем использования интерфейса прикладного программирования (API), допускающего контроль за установочными параметрами, такими как максимальное потребление памяти, время и/или счетчики точек данных.

20 На фиг.1 показана блок-схема операционной среды 100 для обеспечения платформы 105 визуализации. Платформа 105 визуализации может содержать модуль 110 объектов, процессор 115 макета (изображения) и итератор 125 геометрии. Процессор 115 макета может содержать алгоритм 120 пропуска точек данных. Итератор 125 геометрии может содержать алгоритм 130 пропуска геометрии. Итератор 125 геометрии и процессор 115 25 макета могут создавать, обновлять, считывать, извлекать и/или обеспечивать множество данных 135 вектора геометрии. Операционная среда 100 может дополнительно содержать приложение 140, содержащее блок 145 воспроизведения (рендерер), способный воспроизводить (предоставлять?) команды вычерчивания для устройства 150 вывода, такого как монитор, экран, принтер и/или другое устройство отображения. Платформа 30 105 визуализации может содержать архитектуру, позволяющую создание визуализации данных, например диаграммы, и может предоставлять интерактивную возможность визуализации. Архитектура может обеспечивать интеграцию с многими платформами визуального воспроизведения. При выборе пользователем предоставленной возможности архитектура может преобразовывать выборку в обычный формат и 35 модифицировать визуализацию данных в соответствии правилами макета, не зависящими от платформы воспроизведения.

Приложение 140 может задавать визуализацию посредством модуля 110 объектов. Модуль 110 объектов может вызывать процессор 115 макета, который может создать в машинной памяти совокупность записей геометрии в виде данных 135 вектора 40 геометрии. Приложению 140 может далее требоваться воспроизводить визуализацию, и оно может входить в итератор 125 геометрии, который может осуществлять доступ к данным 135 вектора геометрии и возвращать их приложению 140. Приложение 140 может преобразовывать каждый геометрический примитив в данных 135 вектора геометрии в команды вычерчивания для воспроизведения на устройстве 150 вывода.

45 Платформа 105 визуализации может содержать совместно используемые базовые средства («ядро»), содержащие программные библиотеки и/или обслуживающие программы для обеспечения интерактивных визуализаций. Совместно используемые базовые средства могут быть реализованы, например, на языке C++ или C# и могут

быть не зависящими от платформы. Совместно используемые базовые средства могут содержать обслуживающие программы визуализации для обеспечения макетов, форм и/или геометрии, услуг (вычерчивания) линий, 3-мерной визуализации, формирования анимационного кадра и/или интерактивных «горячих» точек. Платформа визуализации 110 может дополнительно содержать интерфейсы прикладного программирования (API) для взаимодействия с приложением 140.

Могут иметься две области для оптимизации производительности в течение составления данных 135 вектора геометрии и преобразования данных 135 вектора геометрии в команды вычерчивания для визуального воспроизведения. Например, в течение составления данных 135 вектора геометрии использование машинной памяти может возрасти значительно, если увеличивается объем данных. Требуемый объем машинной памяти может быть уменьшен с помощью алгоритма 120 пропуска точек данных с тем, чтобы составить векторы геометрии для подмножества данных. Алгоритм 130 пропуска геометрии может действовать для уменьшения числа команд вычерчивания, требуемых для воспроизведения, путем преобразования подмножества данных 135 вектора геометрии.

На фиг.2 показана структурная схема, поясняющая общие этапы, включенные в способ 200 в соответствии с вариантом осуществления изобретения, для обеспечения оптимизации платформы визуализации данных. Способ 200 может быть осуществлен с использованием вычислительного устройства 300, как описано более подробно ниже относительно фиг.3. Пути осуществления этапов способа 200 будут описаны более подробно ниже. Способ 200 может начинаться в начальном блоке 205 и переходить на этап 210, где вычислительное устройство 300 может задавать визуализацию. Например, приложение 140 может посылать на платформу 105 визуализации запрос на создание визуального объекта. Запрос может содержать значение данных, ряд данных и/или тип объекта, такой как линейчатая диаграмма, столбчатая диаграмма, секторная диаграмма, или график. Платформа 105 визуализации может использовать диспетчер объектов, чтобы задавать адрес (размещения в) памяти для визуального объекта.

От этапа 210, где вычислительное устройство 300 задало визуализацию, способ 200 может переходить на этап 220, где вычислительное устройство 300 может составлять, по меньшей мере, один вектор геометрии, связанный с визуализацией. Например, приложение 140 может задать визуализацию диаграммы, содержащей 1000 точек данных, причем каждая точка данных может быть представлена вычерченной восьмиугольной формой. В обычных системах, платформа 105 визуализации может составлять векторы геометрии для каждой из этих 1000 точек данных. В соответствии с вариантами осуществления изобретения процессор 115 макета может использовать алгоритм 120 пропуска точек данных, чтобы составлять векторы геометрии для подмножества этих 1000 точек данных. Это может позволять отображать визуальный объект с качеством черновика и/или предварительного просмотра. Дополнительно в соответствии с вариантами осуществления изобретения процессор 115 макета может действовать, чтобы определять, являются ли точки данных внутри этих 1000 точек данных достаточно близкими друг к другу, чтобы быть неразличимыми, на основании характеристик приложения 140 и/или устройства 150 вывода. Например, если подмножество из 30 точек данных в рамках этих 1000 точек данных все перекрываются или оказываются в пределах заранее определенного порогового значения друг от друга, например в пределах 5 пикселей, алгоритм 120 пропуска точек данных может действовать, чтобы пропустить составление всех из 30 точек данных, и может включать в состав только одну из этих 30 точек данных.

В соответствии с вариантами осуществления изобретения приложение 140 может задать столбчатую диаграмму, содержащую 10000 точек данных. Процессор 115 макета может вызвать класс «последовательность столбцов», который может переходить по данным и составлять векторы для блоков (прямоугольников) для точек данных и помещать их в данные 135 вектора геометрии. Процессор 115 макета может принимать от приложения 140, например, размер диаграммы в пикселах и определять сколько точек данных могут быть отображены в диаграмме этого размера. Например, диаграмма шириной в 300 пикселей может быть способной отображать только 300 точек данных без перекрытия. Процессор макета 135 может таким образом составлять данные вектора для подмножества в 300 точек из этих 10000 точек данных. Размер подмножества может основываться, например, на разрешающей способности устройства 150 вывода.

Размер подмножества также может основываться, например, на определении процессором 115 макета, что некоторые точки данных будут закрываться другими точками данных, и процессор 115 макета может таким образом пропускать составление данных вектора для закрытых точек данных. В соответствии с вариантами осуществления изобретения процессор 115 макета может определить, что более поздняя точка данных занимает точно тот же пиксел, что и предыдущая точка данных, и таким образом более позднюю точку данных можно пропустить. В соответствии с дополнительными вариантами осуществления изобретения алгоритм 120 пропуска точек данных может использовать конфигурируемый критерий, чтобы определять, является ли более поздняя точка достаточно близкой к предыдущей точке данных, подлежащей пропуску. Например, процессор 115 макета может устанавливать критерий, чтобы пропускать точки данных в пределах пяти пикселей относительно предыдущей точки данных. Дополнительно в соответствии с вариантами осуществления изобретения критерий может возбуждаться, если этап составления продолжается, чтобы ограничивать объем машинной памяти, расходуемой составлением данных 135 вектора геометрии. Критерий также может приниматься от приложения 140, например, путем выборки пользователем управляющего элемента «регулятор» (дроссель) пользовательского интерфейса.

Как только вычислительное устройство 300 составит векторы геометрии на этапе 220, способ 200 может переходить на этап 230, где вычислительное устройство 300 может принимать запрос на воспроизведение визуализации. Например, приложение 140 может запрашивать воспроизведение визуализации в соответствии с директивой пользователя.

После того, как вычислительное устройство 300 принимает запрос на визуализацию на этапе 230, способ 200 может переходить на этап 240, где вычислительное устройство 300 может выполнять итерацию по составленным векторам геометрии и преобразовывать их в команды вычерчивания. Например, платформа 105 визуализации может преобразовывать вектор геометрии, содержащий ось X объекта диаграммы, в команду вычерчивания линии, причем команда вычерчивания содержит связанные данные, такие как начальная точка линии, конечная точка линии, цвет линии и/или ширина линии.

В соответствии с вариантами осуществления изобретения платформа 105 визуализации может принимать от приложения, по меньшей мере, одну поддерживаемую команду вычерчивания. Платформа 105 визуализации может действовать, чтобы преобразовывать составленные векторы геометрии в поддерживаемые команды вычерчивания. Например, платформа 105 визуализации может действовать, чтобы линеаризовать вектор геометрии, содержащий восьмиугольник, в команды вычерчивания, содержащие начальную и

конечную точку для каждого набора из восьми линий, используемых для воспроизведения восьмиугольной формы на устройстве 150 вывода. Взамен или дополнительно приложение 140 может информировать платформу визуализации о поддерживаемой команде вычерчивания, содержащей центральную точку и ширину для формы восьмиугольника, так что платформа 105 визуализации может предпочтительнее преобразовать вектор геометрии, содержащий восьмиугольник, в поддерживаемую команду вычерчивания, чем линейизовать форму восьмиугольника. Приложение 140, таким образом, может уменьшить количество преобразуемых команд вычерчивания, которые платформа 105 визуализации должна обеспечить.

Вычислительное устройство 300 может использовать алгоритм 130 пропуска геометрии, чтобы уменьшить количество векторов геометрии, которые необходимо преобразовать и воспроизводить. Например, если визуализация содержит диаграмму из 100 пикселей на 200 пикселей, использующую восьмиугольные формы для точек данных, каждый восьмиугольник может содержать форму из 10 пикселей на 10 пикселей или менее, и некоторые линейные сегменты могут содержать нулевые длины. Алгоритм 130 пропуска геометрии может, например, определить, что конкретный вектор геометрии связан с критерием для пропускаемого, так что вектор может быть преобразован в команду вычерчивания линии, имеющей одну и ту же точку начала и конца. В таком примере, алгоритм 130 пропуска геометрии может предпочтительнее переходить через конкретный вектор геометрии, чем преобразовывать его в команду вычерчивания и посылать его на приложение 140 для воспроизведения. Алгоритм 130 пропуска геометрии может пропускать один из векторов геометрии согласно многим другим критериям, таким как размер отображения, тип данных, тип визуализации, общее число точек данных и общее число векторов геометрии.

В соответствии с вариантами осуществления изобретения приложение 140 может извлекать команды вычерчивания в виде пакетов. Например, приложение 140 может выделить достаточно памяти для приема 50 единиц команд вычерчивания. Платформа 105 визуализации затем может обеспечить 50 команд вычерчивания и информировать приложение 140, ожидаются ли еще команды вычерчивания. Платформа 105 визуализации может ожидать, чтобы приложение 140 запросило следующий пакет, и повторять процесс.

От этапа 240 способ 200 может переходить на этап 250, где вычислительное устройство 300 может воспроизводить визуализацию для устройства вывода. Например, приложение 140 может принимать команды вычерчивания в блок 145 воспроизведения и вычерчивать визуализацию на экране, входящем в состав устройства 150 вывода.

Как только вычислительное устройство 300 воспроизведет визуализацию на этапе 250, способ 200 может переходить на этап 260, где вычислительное устройство 300 может определять, произошло ли изменение в отображении. Например, отображенная визуализация может быть перемещена, изменена в масштабе, изменена в размере и/или скрыта полностью или частично. Если вычислительное устройство определяет, что произошло изменение отображения визуализации, способ 300 может возвращаться на этап 250, где вычислительное устройство 300 может повторно воспроизводить визуализацию. Пропуск точек данных может происходить в течение составления на этапе 220, если данные 135 вектора геометрии помещены в кэш-память, тогда как пропуск геометрии может происходить при всяком воспроизведении визуализации на этапе 250. Так, например, всякий раз, когда приложение 140 должно воспроизводить визуализацию, например, если (область на экране) «окно» перемещается, свертывается в пиктограмму, развертывается на весь экран или изменяется в масштабе, приложение

140 может принимать новый набор команд вычерчивания от итератора 125 геометрии. Если приложение 140, например, уменьшает масштаб изображения, меньше пикселей может отображаться в визуализации, и пропуск геометрии может быть более интенсивным, пропуская больше векторов геометрии.

5 В соответствии с вариантами осуществления изобретения пропуск точек данных и пропуск геометрии можно использовать отдельно и/или вместе и управление может осуществляться по предоставленным пользователем параметрам конфигурации. Например, пользователь, желающий изменить масштаб к большой четкости (детальности) на диаграмме, может запретить пропуск точек данных, чтобы
10 обеспечивать наибольшее количество данных, разрешая при этом пропуск геометрии, чтобы предотвратить преобразование векторов, не видимых в измененном по масштабу состоянии.

Как только вычислительное устройство 300 на этапе 260 определило, произошло ли изменение отображения, и повторно воспроизвело визуализацию, если необходимо,
15 способ 200 может завершиться на этапе 270.

Вариант осуществления в соответствии с изобретением может содержать систему для обеспечения оптимизации платформы визуализации. Система может содержать запоминающее устройство и блок обработки, связанный с запоминающим устройством. Блок обработки может действовать, чтобы задавать визуализацию, создавать множество
20 записей геометрии, принимать запрос на отображение визуализации, осуществлять итерацию по множеству векторов геометрии, преобразовывать, по меньшей мере, одну запись из множества записей геометрии, по меньшей мере, в одну команду вычерчивания, и отображать визуализацию на устройстве отображения.

Другой вариант осуществления в соответствии с изобретением может содержать
25 систему для обеспечения оптимизации платформы визуализации. Система может содержать запоминающее устройство и блок обработки, связанный с запоминающим устройством. Блок обработки может действовать, чтобы создавать объект визуализации, связанный с множеством значений данных, составлять, по меньшей мере, одно первое значение из множества значений данных в вектор геометрии, связанный с объектом
30 визуализации, и пропускать составляющее, по меньшей мере, одно второе значение из множества значений данных.

Еще один вариант осуществления в соответствии с изобретением может содержать систему для обеспечения оптимизации платформы визуализации. Система может
35 содержать запоминающее устройство и блок обработки, связанный с запоминающим устройством. Блок обработки может действовать для приема от пользовательского приложения запроса на создание объекта визуализации, составления набора векторов геометрии, связанных с объектом визуализации, приема от пользовательского приложения запроса на воспроизведение, преобразования каждого подмножества из
40 набора векторов геометрии, по меньшей мере, в одну команду вычерчивания, послышки, по меньшей мере, одной команды вычерчивания, связанной с каждым подмножеством из набора векторов геометрии, на пользовательское приложение и воспроизведения объекта визуализации. Блок обработки может дополнительно действовать для приема от пользовательского приложения второго запроса на воспроизведение в ответ на
45 обнаруженное изменение, воздействующее на воспроизведенный объект визуализации, преобразования каждого второго подмножества из набора векторов геометрии, по меньшей мере, в одну команду вычерчивания, послышки, по меньшей мере, одной команды вычерчивания, связанной с каждым вторым подмножеством из набора векторов геометрии, на пользовательское приложение, и повторного воспроизведения

объекта визуализации.

На фиг.3 показана блок-схема системы, включающей в состав вычислительное устройство 300. В соответствии с вариантом осуществления изобретения вышеупомянутые запоминающее устройство и блок обработки могут быть реализованы в вычислительном устройстве, таком как вычислительное устройство 300 по фиг.3. Любая подходящая комбинация аппаратных средств, программного обеспечения или микропрограммного обеспечения может использоваться для реализации запоминающего устройства и блока обработки. Например, запоминающее устройство и блок обработки могут быть реализованы с помощью вычислительного устройства 300 или любого из других вычислительных устройств 318 в комбинации с вычислительным устройством 300. Вышеупомянутая система, устройство и процессоры являются примерами, и другие системы, устройства и процессоры могут содержать вышеупомянутые запоминающее устройство и блок обработки в соответствии с вариантами осуществления изобретения. Кроме того, вычислительное устройство 300 может содержать операционную среду для системы 100, как описано выше. Система 100 может работать в других средах и не ограничивается вычислительным устройством 300.

Со ссылкой на фиг.3 система в соответствии с вариантом осуществления изобретения может включать в себя вычислительное устройство, такое как вычислительное устройство 300. В основной конфигурации, вычислительное устройство 300 может включать в себя, по меньшей мере, один блок 302 обработки и системную память 304. В зависимости от конфигурации и типа вычислительного устройства системная память 304 может содержать, но не ограничивается указанными, энергозависимую (например, оперативное запоминающее устройство (ОЗУ, RAM)), энергонезависимую (например, постоянное запоминающее устройство (ПЗУ, ROM)), флэш-память или какую-либо комбинацию. Системная память 304 может включать в себя операционную систему 305, один или несколько программных модулей 306 и может включать в себя платформу визуализации 110. Операционная система 305, например, может быть подходящей для операции управления вычислительным устройством 300. В одном варианте осуществления программные модули 306 могут включать в себя пользовательское приложение 140. Кроме того, варианты осуществления изобретения также могут быть реализованы на практике вместе с графической библиотекой, другими операционными системами или любой другой прикладной программой, и не ограничиваются каким-либо конкретным приложением или системой. Эта основная конфигурация проиллюстрирована на фиг.3 этими компонентами внутри пунктирной линии 308.

Вычислительное устройство 300 может иметь дополнительные характеристики или функциональные возможности. Например, вычислительное устройство 300 может также включать в себя дополнительные устройства хранения данных (съёмные и/или несъёмные) такие как, например, ЗУ на магнитных дисках, оптических дисках, или на магнитной ленте. Такое дополнительное ЗУ проиллюстрировано на фиг.3 съёмным ЗУ 309 и несъёмным ЗУ 310. Носители данных (запоминающие среды) компьютера могут включать в себя энергозависимые и энергонезависимые, съёмные и несъёмные носители, реализованные любым способом или технологией, предназначенным для хранения информации, такой как машиночитаемые команды, структуры данных, программные модули, или других данных. Системная память 304, съёмное ЗУ 309 и несъёмное ЗУ 310 все являются примерами носителей данных компьютера (то есть запоминающих устройств). Носители данных компьютера могут включать в себя, но не ограничиваются указанными, ОЗУ, ПЗУ, электрически стираемое ПЗУ (EEPROM), флэш-память или другую технологию памяти, ПЗУ на компакт-диске (CD-ROM), цифровые

многофункциональные диски (DVD) или другое оптическое ЗУ, ЗУ на магнитной кассете, магнитной ленте, магнитных дисках или другие магнитные устройства хранения, или любой другой носитель, который может использоваться для хранения информации и к которому может осуществлять доступ вычислительное устройство 300. Любые такие носители данных компьютера могут быть частью устройства 300. Вычислительное устройство 300 может также иметь в составе устройство(а) 312 ввода, такое как клавиатура, мышь, перо, устройство звукового ввода, устройство сенсорного ввода и т.д. Устройство(а) 314 вывода, такое как устройство отображения, динамики, принтер, и т.д., также могут включаться в состав. Вышеупомянутые устройства являются примерами и могут использоваться другие.

Вычислительное устройство 300 также может содержать соединение 316 связи, которое может позволять устройству 300 осуществлять связь с другими вычислительными устройствами 318, например, по сети в распределенной вычислительной среде, например, внутрикорпоративной или сети Интернет. Соединение 316 связи является одним примером сред (носителей) передачи данных. Среды передачи данных обычно могут составлять машиночитаемые команды, структуры данных, программные модули или другие данные в виде модулированного сигнала данных, такого как несущая или другой транспортный механизм, и включают в себя любые среды для доставки информации. Термин «модулированный сигнал данных» может означать сигнал, у которого одна или несколько характеристик установлены или изменяются таким образом, чтобы закодировать информацию в сигнале. В качестве примера, а не ограничения, среды передачи данных могут включать в себя проводные среды, такие как проводная сеть или непосредственное проводное соединение, и беспроводные среды, такие как акустические, радиочастотные (РЧ, RF), инфракрасные и другие беспроводные среды передачи данных. Термин «читаемый компьютером носитель», как используется в документе, может включать в себя и компьютерный носитель данных, и среду передачи данных.

Как изложено выше, ряд программных модулей и файлов данных может храниться в системной памяти 304, включая операционную систему 305. При исполнении в блоке 302 обработки программные модули 306 (например, пользовательское приложение 140) могут выполнять процессы, включая, например, один или несколько этапов способов 200, как описано выше. Вышеупомянутый процесс является примером, и блок 302 обработки может выполнять другие процессы. Другие программные модули, которые могут использоваться в соответствии с вариантами осуществления настоящего изобретения, могут включать в себя приложения электронной почты и контактов, приложения обработки текстов, приложения электронных таблиц, приложения баз данных, приложения представления слайдов, графические или автоматизированные прикладные программы и т.д.

В целом, в соответствии с вариантами осуществления изобретения программные модули могут включать в себя подпрограммы, программы, компоненты, структуры данных и другие типы структур, которые могут выполнять конкретные задачи или которые могут реализовывать особые абстрактные типы данных. Кроме того, варианты осуществления изобретения могут быть реализованы с помощью других конфигураций вычислительных систем, включая ручные устройства, мультипроцессорные системы, или микропроцессорную программируемую бытовую электронику, миникомпьютеры, универсальные компьютеры и т.п. Варианты осуществления изобретения могут также быть реализованы в распределенных вычислительных средах, где задачи выполняются устройствами удаленной обработки, которые связаны через сеть связи. В распределенной

вычислительной среде программные модули могут находиться и в локальных, и удаленных устройствах хранения данных.

Кроме того, варианты осуществления изобретения могут быть реализованы в виде электрической схемы, содержащей дискретные электронные элементы, пакетированные или интегральные электронные микросхемы, содержащие логические вентили, схемы, использующей микропроцессор, или на отдельном кристалле, содержащем электронные элементы или микропроцессоры. Варианты осуществления изобретения также могут быть реализованы с использованием других технологий, способных выполнять логические операции, такие как, например, AND (И), OR (ИЛИ), и NOT (НЕ), включающих в себя без ограничения указанными, механические, оптические, струйные и квантовые технологии. Кроме того, варианты осуществления изобретения также могут быть реализованы в рамках универсального компьютера или в любых других схемах или системах.

Варианты осуществления изобретения, например, могут быть реализованы в виде вычислительного процесса (способа), вычислительной системы или в виде изделия, такого как компьютерный программный продукт или читаемый компьютером носитель. Компьютерным программным продуктом может быть компьютерный носитель данных, читаемый вычислительной системой и с кодированием команд компьютерной программы для исполнения вычислительного процесса. Компьютерным программным продуктом может также быть распространяемый сигнал на несущей, считываемый вычислительной системой и с кодированием компьютерной программы в командах для исполнения вычислительного процесса. Соответственно, настоящее изобретение может быть реализовано в виде аппаратных средств и/или программного обеспечения (включая микропрограммное обеспечение, резидентное программное обеспечение, микрокод и т.д.). Другими словами, варианты осуществления настоящего изобретения могут иметь форму компьютерного программного продукта на используемом компьютером или читаемом компьютером носителе данных, содержащем используемый компьютером или читаемый компьютером код программы, реализованный в носителе, для использования системой исполнения команд или в связи с таковой. Используемый компьютером или читаемый компьютером носитель может быть любым носителем, который может содержать, хранить, передавать, распространять или транспортировать программу для использования системой исполнения команд, аппаратурой или устройством, или в связи с таковыми.

Используемым компьютером или читаемым компьютером носителем может быть, например, без ограничения указанными, электронная, магнитная, оптическая, электромагнитная, инфракрасная или полупроводниковая система, аппаратура, устройство, или среда распространения. Для более конкретных примеров читаемых компьютером носителей (неисчерпывающий перечень), читаемый компьютером носитель может включать в себя нижеследующее: электрическое соединение с наличием одного или нескольких проводов, портативную (съёмную) дискету компьютера, оперативное запоминающее устройство (ОЗУ), постоянное запоминающее устройство (ПЗУ), стираемое программируемое ПЗУ (EPROM или флэш-память), светопровод и портативное ПЗУ на компакт-диске (CD-ROM). Обратите внимание, что используемый компьютером или читаемый компьютером носитель может быть даже бумажным или другим подходящим носителем, на котором напечатана программа, если программу можно электронным образом вводить с оцифровкой посредством, например, оптического сканирования бумажного или другого носителя, затем компилировать, интерпретировать или иначе обрабатывать подходящим образом в случае

необходимости, и затем сохранять в машинной памяти.

Варианты осуществления настоящего изобретения, например, описаны выше со ссылкой на блок-схемы и/или операционные иллюстрации способов, систем и компьютерных программных продуктов согласно вариантам осуществления изобретения. Функции/действия, отмеченные на блоках схемы, могут происходить вне порядка следования, как показано на любой структурной схеме. Например, два этапа, показанные подряд, фактически могут выполняться по существу одновременно, или этапы могут иногда выполняться в обратном порядке следования в зависимости от рассматриваемой функциональности/действий.

Тогда как были описаны некоторые варианты осуществления изобретения, могут иметься другие варианты осуществления. Кроме того, хотя варианты осуществления настоящего изобретения были описаны в виде связываемых с данными, хранимыми в машинной памяти и других запоминающих средах, данные также могут храниться на других типах читаемых компьютером носителей или считываться с таковых, например, внешних запоминающих устройств, таких как накопители на жестких дисках, гибких дисках, или CD-ROM, сигнал несущей из сети Интернет, или других формах ОЗУ или ПЗУ. Дополнительно, этапы раскрытых способов могут модифицироваться любым образом, включая изменение порядка следования этапов и/или вставку или удаление этапов, без выхода за рамки изобретения.

Все права, включая авторские права на код, включенный в документ, принадлежат и являются собственностью Заявителя. Заявитель закрепляет и сохраняет все права на код, включенный в документ, и предоставляет разрешение на переиздание материала только в связи с переизданием выданного патента и ни для какой другой цели.

Тогда как описание включает примеры, объем изобретения указан нижеследующей формулой изобретения. Кроме того, хотя описание было выполнено на языке, конкретно определенном для структурных признаков и/или методологических действий, формула изобретения не ограничивается признаками или действиями, описанными выше. Предпочтительнее, конкретные признаки и действия, описанные выше, раскрыты в качестве примеров для реализаций изобретения.

30

Формула изобретения

1. Способ выполнения визуализации посредством платформы визуализации данных, причем способ содержит:

- 35 задание визуализации;
- создание множества векторов геометрии;
- прием запроса на отображение визуализации;
- выполнение итерации по множеству векторов геометрии;
- пропуск, по меньшей мере, одного первого вектора из множества геометрических записей согласно алгоритму пропуска геометрии, содержащему, по меньшей мере, один из следующих критериев: обнаружение линии нулевой длины, общее число множества векторов геометрии, общее число точек данных, размер отображения, тип данных и тип визуализации;
- 40 преобразование, по меньшей мере, одного второго вектора из множества записей геометрии, по меньшей мере, в одну команду вычерчивания; и
- 45 отображение визуализации на устройстве отображения.

2. Способ по п.1, дополнительно содержащий:

определение, содержит ли, по меньшей мере, один второй вектор из множества векторов геометрии местоположение пиксела, связанное, по меньшей мере, с одним

третьим вектором из множества векторов геометрии; и

в ответ на определение того, что, по меньшей мере, один второй вектор из множества векторов геометрии содержит местоположение пиксела, связанное, по меньшей мере, с одним вторым вектором множества векторов геометрии, выполнение пропуска, по меньшей мере, одного второго вектора из множества векторов геометрии.

3. Способ по п.1, дополнительно содержащий:

посылку множества преобразованных векторов геометрии в приложение; и воспроизведение множества преобразованных векторов геометрии для отображения посредством приложения.

4. Способ по п.3, дополнительно содержащий:

прием от приложения запроса на пакет преобразованных векторов геометрии; посылку подмножества из множества преобразованных векторов геометрии в приложение, причем подмножество из множества преобразованных векторов геометрии содержит некоторое количество преобразованных записей геометрии, связанных с принятым запросом пакета; и

ожидание, чтобы послать остающееся подмножество из множества преобразованных векторов геометрии.

5. Способ по п.1, в котором преобразование, по меньшей мере, одного из множества векторов геометрии, по меньшей мере, в одну команду вычерчивания содержит линеаризацию, по меньшей мере, одного из множества векторов геометрии.

6. Способ по п.1, дополнительно содержащий:

прием, по меньшей мере, одной поддерживаемой команды вычерчивания, причем преобразование, по меньшей мере, одного из множества векторов геометрии, по меньшей мере, в одну команду вычерчивания содержит преобразование, по меньшей мере, одного из множества векторов геометрии, по меньшей мере, в одну поддерживаемую команду вычерчивания.

7. Система для выполнения визуализации посредством платформы визуализации данных, причем система содержит:

запоминающее устройство; и

блок обработки, связанный с запоминающим устройством, причем блок обработки выполнен с возможностью:

создания объекта визуализации, связанного с множеством значений данных, включения, по меньшей мере, одного первого значения из множества значений данных в вектор геометрии, связанный с объектом визуализации, и

пропуска составляющего, по меньшей мере, одного второго значения из множества значений данных согласно предварительно заданному критерию, и

преобразования, по меньшей мере, подмножества из множества векторов геометрии в множество команд вычерчивания, причем подмножество из множества векторов геометрии выбрано согласно алгоритму пропуска геометрии, содержащему, по меньшей мере, один из следующих критериев: обнаружение линии нулевой длины, общее число множества векторов геометрии, общее число точек данных, размер отображения, тип данных и тип визуализации.

8. Система по п.7, причем предварительно заданный критерий является конфигурируемым пользователем.

9. Система по п.7, причем алгоритм пропуска геометрии дополнительно выполнен с возможностью:

преобразования, по меньшей мере, одного первого вектора из множества векторов геометрии, по меньшей мере, в одну команду вычерчивания,

выполнения итерации, по меньшей мере, по одному второму вектору из множества векторов геометрии;

определения, включает ли, по меньшей мере, один второй вектор из множества векторов геометрии точку на объекте визуализации, ближайшую к точке на объекте визуализации, связанном, по меньшей мере, с одним первым вектором; и

в ответ на определение того, что, по меньшей мере, один второй вектор из множества векторов геометрии содержит точку на объекте визуализации, ближайшую к точке на объекте визуализации, связанном, по меньшей мере, с одним первым вектором, выполнение пропуска преобразования, по меньшей мере, одного второго вектора, по меньшей мере, в одну команду вычерчивания.

10. Система по п.7, дополнительно выполненная с возможностью:

приема, по меньшей мере, одного изменения в объект визуализации; и

в ответ на прием, по меньшей мере, одного изменения в объект визуализации, выполнения повторного воспроизведения объекта визуализации.

11. Система по п.10, причем, по меньшей мере, одно изменение в объект визуализации содержит, по меньшей мере, одно из следующего: изменение размера объекта визуализации, увеличение масштаба объекта визуализации, уменьшение масштаба объекта визуализации, перемещение объекта визуализации, частичное закрытие объекта визуализации и перемещение приложения.

12. Система по п.7, дополнительно содержащая пользовательское приложение, выполненное с возможностью воспроизводить объект визуализации на устройство отображения.

13. Читаемый компьютером носитель, который хранит набор команд, который при исполнении выполняет способ выполнения визуализации посредством платформы визуализации данных, причем способ исполняемый посредством набора команд, содержит:

прием от пользовательского приложения запроса на создание объекта визуализации, причем запрос содержит множество значений данных и тип визуализации;

составление множества векторов геометрии, связанных с объектом визуализации, причем составление множества векторов геометрии содержит:

составление, по меньшей мере, одного вектора геометрии, связанного, по меньшей мере, с одним значением из множества значений данных, и

пропуск, по меньшей мере, одного значения из множества значений данных;

прием запроса на воспроизведение от пользовательского приложения;

преобразование каждого подмножества из множества векторов геометрии, по меньшей мере, в одну команду вычерчивания, причем подмножество из множества векторов геометрии выбирается в соответствии с алгоритмом пропуска геометрии;

посылку, по меньшей мере, одной команды вычерчивания, связанной с каждым подмножеством из множества векторов геометрии, в пользовательское приложение;

воспроизведение объекта визуализации, причем воспроизведение объекта визуализации содержит воспроизведение, по меньшей мере, одной команды вычерчивания, связанной с каждым подмножеством из множества векторов геометрии, для устройства отображения;

прием второго запроса на воспроизведение от пользовательского приложения в ответ на обнаруженное изменение, воздействующее на воспроизводимый объект визуализации;

преобразование каждого из второго подмножества из множества векторов геометрии, по меньшей мере, в одну команду вычерчивания;

посылку, по меньшей мере, одной команды вычерчивания, связанной с каждым вторым подмножеством из множества векторов геометрии, в пользовательское приложение; и
повторное воспроизведение объекта визуализации.

5

10

15

20

25

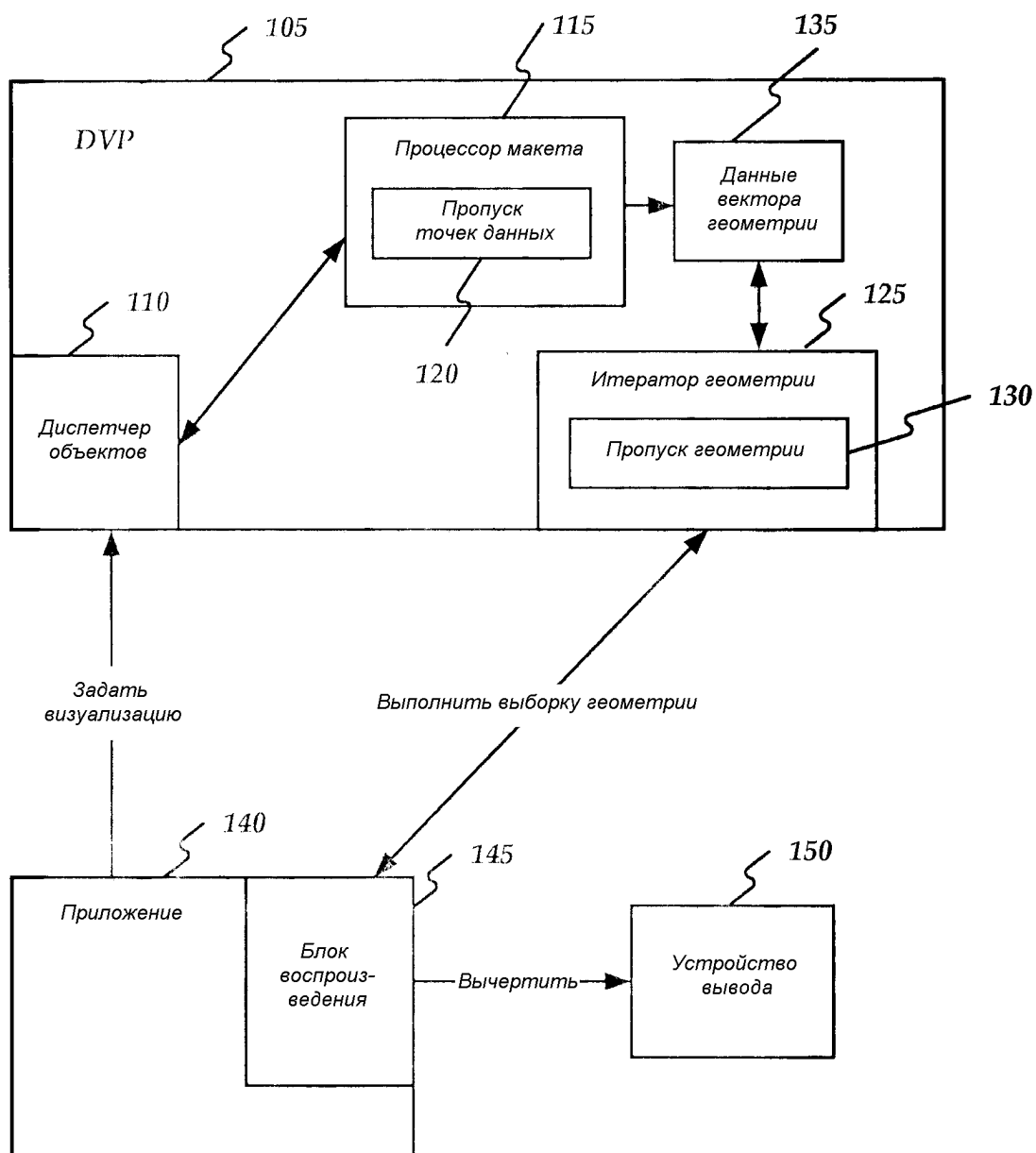
30

35

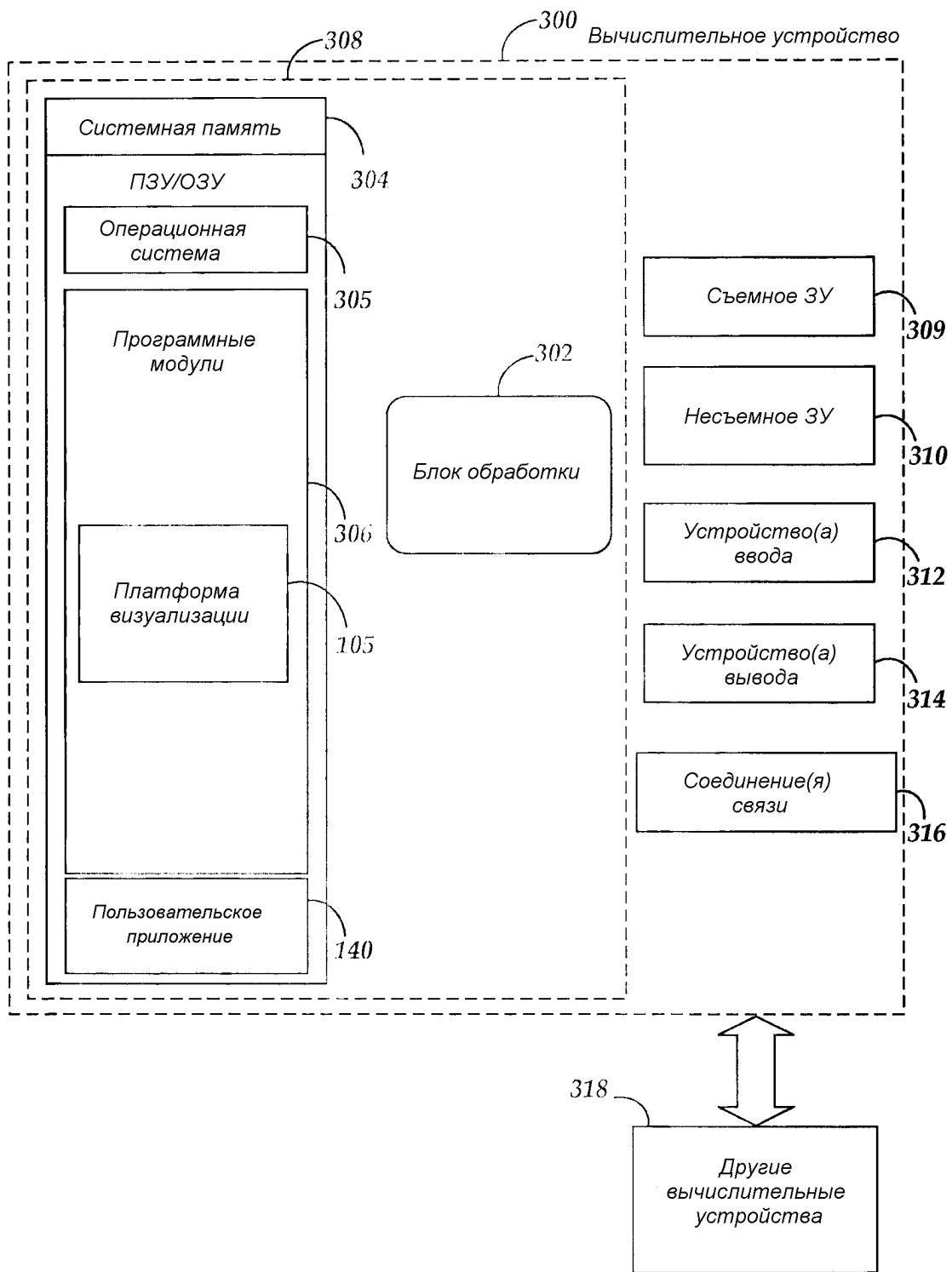
40

45

100



Фиг.1



Фиг.3