



(51) МПК  
*G06K 9/36* (2006.01)  
*G06K 9/46* (2006.01)  
*G06T 5/30* (2006.01)  
*G06T 7/246* (2017.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА  
 ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2016129076, 15.07.2016

(24) Дата начала отсчета срока действия патента:  
 15.07.2016

Дата регистрации:  
 15.08.2017

Приоритет(ы):

(22) Дата подачи заявки: 15.07.2016

(45) Опубликовано: 15.08.2017 Бюл. № 23

Адрес для переписки:  
 127273, Москва, а/я 56, ООО "Аби Девелопмент",  
 Марей Сергей Владимирович

(72) Автор(ы):

Качер Ольга Арнольдовна (RU),  
 Загайнов Иван Германович (RU),  
 Логинов Василий Васильевич (RU)

(73) Патентообладатель(и):

Общество с ограниченной ответственностью  
 "Аби Девелопмент" (RU)

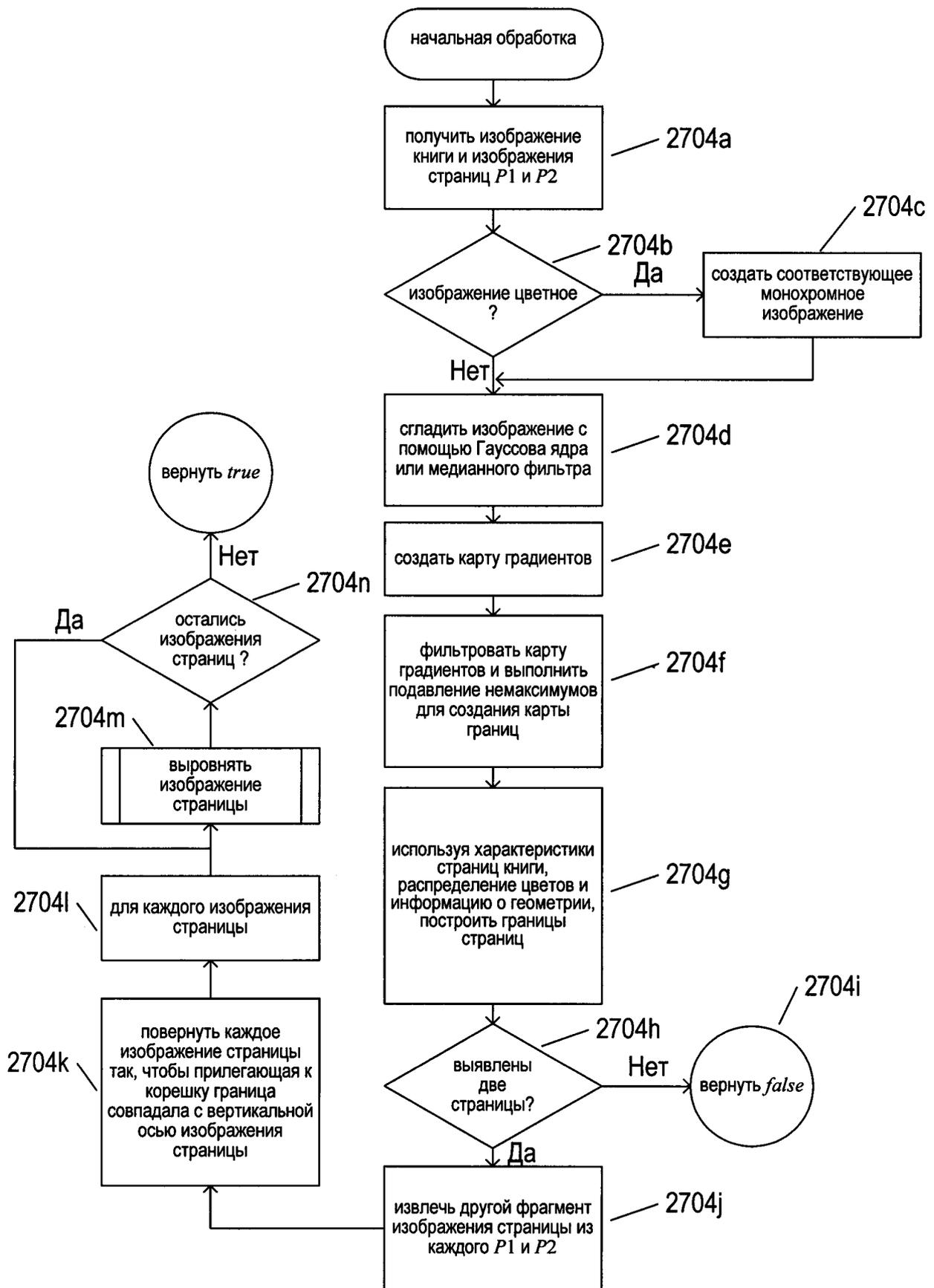
(56) Список документов, цитированных в отчете  
 о поиске: статья ВНАGYASHREE JAY  
 SHAN "CORRECTION OF CURVED TEXT  
 LINES FROM DOCUMENT IMAGE USING  
 PIXEL MANIPULATION", 2015. статья Т.  
 KASAR "ALIGNMENT OF CURVED TEXT  
 STRINGS FOR ENHANCED OCR  
 READABILITY", 2013. US 2010/014782 A1,  
 21.01.2010. US 2013/039598 A1, 14.02.2013. WO  
 02/25575 A2, 28.03.2002. статья  
 "УКРОЩЕНИЕ СТРОПТИВОГО (НА  
 САМОМ ДЕЛЕ, НЕТ) FINE (см. прод.)

(54) СПОСОБ И СИСТЕМА ПОДГОТОВКИ СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
 ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ

(57) Реферат:

Группа изобретений относится к технологиям обработки изображений и оптическому распознаванию символов. Техническим результатом является расширение арсенала технических средств систем оптического распознавания текста. Предложена подсистема обработки изображений в составе устройства, прибора или системы, предназначенных для оптического распознавания текста. Подсистема получает содержащее текст с изогнутыми текстовыми строками изображение и вводит полученное содержащее текст изображение в подсистему обработки изображений, которая, в свою очередь, формирует соответствующее исправленное изображение с выпрямленными текстовыми строками с прямоугольной

организацией. При этом такая подсистема обработки изображений включает отдельный аппаратный уровень или аппаратный уровень, используемый совместно с устройством, прибором или системой, включающий один или более процессоров, один или более модулей памяти. Подсистема также содержит команды в машинном коде, хранящиеся в одном или более физических устройств хранения данных, которые при их выполнении одним или несколькими из одного или более процессоров управляют работой подсистемы обработки изображений в части следующих операций: получение введенного содержащего текст изображения; определение на содержащем текст изображении подизображения страницы. 3 н. и 19 з.п. ф-лы, 77 ил.



ФИГ. 27В

(56) (продолжение):  
 READER", 30.10.2014. RU 2557461 C2, 20.07.2015.



FEDERAL SERVICE  
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.  
*G06K 9/36* (2006.01)  
*G06K 9/46* (2006.01)  
*G06T 5/30* (2006.01)  
*G06T 7/246* (2017.01)

(12) **ABSTRACT OF INVENTION**

(21)(22) Application: **2016129076, 15.07.2016**

(24) Effective date for property rights:  
**15.07.2016**

Registration date:  
**15.08.2017**

Priority:  
(22) Date of filing: **15.07.2016**

(45) Date of publication: **15.08.2017** Bull. № 23

Mail address:  
**127273, Moskva, a/ya 56, OOO "Abi Development",  
Marej Sergej Vladimirovich**

(72) Inventor(s):  
**Kacher Olga Arnoldovna (RU),  
Zagaynov Ivan Germanovich (RU),  
Loginov Vasily Vasilyevich (RU)**

(73) Proprietor(s):  
**Obshchestvo s ogranichennoj otvetstvennostyu  
"Abi Development" (RU)**

(54) **METHOD AND SYSTEM OF PREPARING TEXT-CONTAINING IMAGES TO OPTICAL RECOGNITION OF SYMBOLS**

(57) Abstract:

FIELD: physics.

SUBSTANCE: image processing subsystem is proposed within a device, the device or system designed for optical text recognition. The subsystem receives an image containing text with the curved strings and enters the resulting text-containing image into the image processing subsystem, which in turn generates a corresponding corrected image with rectified text strings with a rectangular arrangement. Herewith such image processing subsystem includes a separate hardware layer or a hardware layer used in conjunction with a device, an apparatus, or a system, including one or more

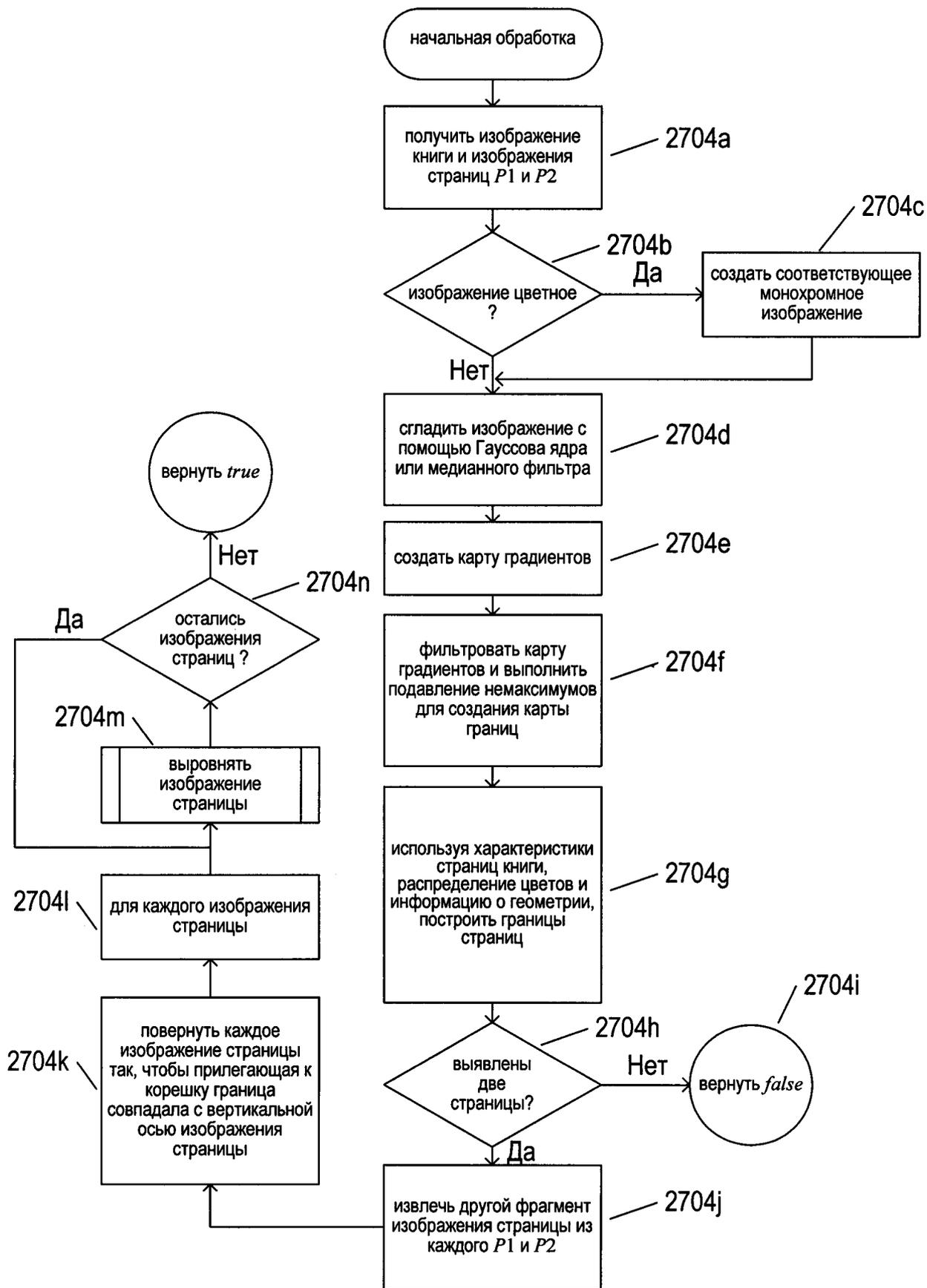
processors, one or more memory modules. The subsystem also contains instructions in the machine code stored in one or more physical storage devices that, when executed by one or more of one or more processors, control the operation of the image processing subsystem in part of the following operations: obtaining the input text containing the text; defining a sub-image of the page on the text containing image.

EFFECT: expanding the arsenal of technical means of optical text recognizing systems.

22 cl, 77 dwg

**RU 2 628 266 C 1**

**RU 2 628 266 C 1**



ФИГ. 27В

## ОБЛАСТЬ ТЕХНИКИ

Настоящий документ относится к обработке изображений и оптическому распознаванию символов и, в частности, к способу и системе, позволяющим выпрямлять искривления содержащих текст изображений, вызванные изогнутой поверхностью 5 страницы и искажениями перспективы, при подготовке к применению способов автоматизированного оптического распознавания символов на содержащих текст изображениях.

## УРОВЕНЬ ТЕХНИКИ

Печатные документы на естественном языке до сих пор являются широко 10 распространенным средством, используемым для коммуникации между людьми, в рамках организаций, а также для распространения информации среди ее потребителей. С появлением повсеместно используемых мощных вычислительных ресурсов, включая персональные вычислительные ресурсы, реализованные в смартфонах, планшетах, ноутбуках и персональных компьютерах, а также с распространением более мощных 15 вычислительных ресурсов облачных вычислительных сервисов, центров обработки данных и корпоративных серверов организаций и предприятий, шифрование и обмен информацией на естественном языке все более часто выполняется в виде электронных документов. В отличие от печатных документов, которые по своей сути представляют собой изображения, электронные документы содержат последовательности цифровых 20 кодов символов и знаков естественного языка. Поскольку электронные документы имеют перед печатными документами преимущества по стоимости, возможностям передачи и рассылки, простоте редактирования и изменения, а также по надежности хранения, за последние 50 лет развилась целая отрасль, поддерживающая способы и системы преобразования печатных документов в электронные. Вычислительные способы 25 и системы оптического распознавания символов совместно с электронными сканерами являются надежными и экономичными средствами получения изображений печатных документов и компьютерной обработки получаемых цифровых изображений содержащих текст документов с целью создания электронных документов, соответствующих печатным.

30 Раньше электронные сканеры представляли собой крупногабаритные настольные или напольные электронные устройства. Однако, с появлением смартфонов, оснащенных камерами, а также других мобильных устройств получения изображения с процессорным управлением появилась возможность получения цифровых изображений содержащих текст документов с помощью широкого диапазона различных типов широко 35 распространенных портативных устройств, включая смартфоны, недорогие цифровые камеры, недорогие камеры видеонаблюдения, а также устройства получения изображений в мобильных вычислительных приборах, включая планшетные компьютеры и ноутбуки. Цифровые изображения содержащих текст документов, создаваемые этими портативными устройствами и приборами, могут обрабатываться 40 вычислительными системами оптического распознавания символов, в том числе приложениями оптического распознавания символов в смартфонах для создания соответствующих электронных документов.

К сожалению, содержащие текст изображения, полученные с помощью портативных средств получения изображений документов, часто бывают искажены наличием шума, 45 оптического размытия, искривлений линейных текстовых строк, вызванных изогнутой поверхностью страниц и искажением перспективы, а также других дефектов и недостатков. Даже изображения, полученные с помощью специальных устройств сканирования документов, могут иметь искривления линейных текстовых строк,

вызванные искажением перспективы, когда книга при сканировании раскрывается и кладется лицевой стороной вниз на прозрачную сканирующую поверхность. Эти дефекты и недостатки могут значительно снизить производительность вычислительного оптического распознавания символов и значительно увеличить частоту ошибочного распознавания символов и неспособности способов и систем оптического распознавания символов обеспечить точное кодирование текста, содержащегося на цифровых изображениях. По этой причине проектировщики и разработчики устройств и приборов получения изображений и способов и систем оптического распознавания символов, а также пользователи данных устройств, приборов и систем оптического распознавания символов продолжают искать способы и системы, позволяющие устранить дефекты и недостатки, присущие многим содержащим текст цифровым изображениям, включая содержащие текст цифровые изображения, полученные с помощью мобильных устройств, которые затрудняют дальнейшую вычислительную обработку содержащих текст цифровых изображений.

### РАСКРЫТИЕ ИЗОБРЕТЕНИЯ

Настоящий документ относится к способам и системам выпрямления искривления текстовых строк содержащих текст цифровых изображений, включая содержащие текст цифровые изображения, полученные из двух страниц раскрытой книги. При первоначальной обработке содержащего текст изображения определяется контур содержащей текст страницы. Затем в рамках изображения страницы с обозначенным контуром определяются скопления символов, в том числе слова и фрагменты слов. Определяются центры и углы наклона скоплений символов, так чтобы каждое скопление символов было вписано в прямоугольник наименьших возможных размеров, ориентированный в соответствии с углом наклона, определенным для описываемого скопления символов. Для перспективного искажения на содержащем текст изображении исходя из описанных скоплений символов строится модель, которая затем уточняется с помощью дополнительной информации, извлеченной из содержащего текст изображения. Данная модель, которая, по сути, является картой углов наклона, позволяет присваивать пикселям на изображении страницы значения локального смещения, которые затем используются для выпрямления строк текста на содержащем текст изображении.

### КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

На Фиг. 1 показана архитектурная схема вычислительной системы верхнего уровня, такой как вычислительная система, в которой для исправления искривления текстовых строк, вызванного искажением перспективы, применяется способ настоящего изобретения.

На Фиг. 2A-D показаны два различных типа портативных устройств получения изображения.

На Фиг. 3 показано типовое изображение с цифровым кодированием.

На Фиг. 4 показан один из вариантов цветовой модели RGB.

На Фиг. 5 показана другая цветовая модель «оттенок-насыщенность-светлота» (HSL).

На Фиг. 6 изображено формирование полутонового или бинарного изображения из цветного.

На Фиг. 7 показано дискретное вычисление градиента яркости.

На Фиг. 8 показан градиент, рассчитанный для некой точки на непрерывной поверхности.

На Фиг. 9 показан ряд примеров градиента яркости.

На Фиг. 10 показано применение ядра к изображению.

На Фиг. 11 показана свертка ядра с изображением с получением преобразованного изображения.

На Фиг. 12 показан пример ядра и методик обработки изображений на основе ядра.

5 На Фиг. 13А-В показан один вариант реализации способа, к которому относится настоящий документ.

На Фиг. 14А-С показана первоначальная обработка изображения страницы с целью подготовки к выравниванию вертикальных границ страницы, как описано выше со ссылкой на Фиг. 13В.

10 На Фиг. 15А-В показан один подход к выравниванию границ страницы, который описан выше со ссылкой на Фиг. 13В.

На Фиг. 16А-Г показано разделение фрагментов на блоки.

На Фиг. 17А-Д показан процесс разделения фрагментов на блоки.

15 На Фиг. 18А-В показан первый этап фильтрации, примененный к начальным блокам фрагментов, полученным методом разделения фрагментов на блоки, описанным со ссылкой на Фиг. 17А-Д.

На Фиг. 19А-В показано формирование начальной модели искривления текстовых строк на изображении страницы.

На Фиг. 20 приведена блок-схема вторичной фильтрации блоков фрагментов.

20 На Фиг. 21А-Г показаны операции дилатации, эрозии, замыкания и открытия, используемые в описываемом варианте реализации способов настоящего изобретения для формирования закрашенных контуров, соответствующих текстовым строкам на содержащем текст изображении.

25 На Фиг. 22 показана операция замыкания с последующим открытием, выполняемая в отношении содержащего текст изображения для получения закрашенных контуров, соответствующих строкам текста.

На Фиг. 23А-Д показано формирование сегментов изогнутой текстовой строки из центроидов блоков фрагментов.

30 На Фиг. 24А-В показано построение части сегмента на текущем конце изогнутого сегмента текстовой строки.

На Фиг. 25 показаны особенно сильно искривленные области изображений страниц и вычисление коэффициентов корректировки.

35 На Фиг. 26А-Г показано выпрямление изогнутых текстовых строк на изображении страницы для получения прямолинейных текстовых строк с использованием карты углов наклона, в которой с каждым пикселем на изображении страницы соотнесен некий угол наклона.

40 На Фиг. 27А-С представлены блок-схемы, иллюстрирующие один из вариантов реализации способа корректировки перспективного искажения текстовых строк на изображениях текста, снятых мобильными устройствами и другими видами оборудования и приборов получения изображения текста.

#### ПОДРОБНОЕ ОПИСАНИЕ

Настоящий документ относится к способам и системам, позволяющим выпрямлять искривленные строки текста на содержащих текст изображениях при подготовке содержащих текст изображений к точному оптическому распознаванию символов. В первом подразделе ниже со ссылкой на Фиг. 1-12 приводится краткое введение в архитектуру вычислительной системы, цифровые изображения и способы обработки цифровых изображений. Во втором подразделе со ссылкой на Фиг. 13А-26F приводится подробное описание способов и систем настоящего изобретения. В заключительном

подразделе со ссылкой на Фиг. 27A-S с помощью блок-схем иллюстрируется один из вариантов реализации способа настоящего изобретения.

Обзор архитектуры вычислительной системы, цифровых изображений и способов обработки цифровых изображений

5 На Фиг. 1 показана схема архитектуры вычислительной системы верхнего уровня, такой как вычислительная система, в которой для исправления искривления текстовой строки, вызванного искажением перспективы, применяется способ настоящего изобретения. Мобильные устройства получения изображений, включая смартфоны и цифровые камеры, могут быть представлены схематически аналогичным образом и  
10 также могут содержать процессоры, память и внутренние шины. Тем, кто знаком с современной наукой и технологиями, будет понятно, что программы управления или подпрограмма управления, включающая команды в машинном коде, которые хранятся в физической памяти устройства с процессорным управлением, представляют собой компонент управления данным устройством и являются настолько же физическими,  
15 реальными и важными, насколько и любой другой компонент электромеханического устройства, включая устройства получения изображений. Компьютерная система содержит один или более центральных процессоров (ЦП) 102-105, один или более электронных модулей памяти 108, взаимосвязанных с ЦП через шину подсистемы ЦП/память 110 или несколько шин, первый мост 112, который соединяет шину подсистемы  
20 ЦП/память 110 с дополнительными шинами 114 и 116, либо другие виды средств высокоскоростного соединения, в том числе множественные высокоскоростные последовательные соединения. Данные шины или последовательные межсоединения в свою очередь соединяют ЦП и память со специализированными процессорами, такими как графический процессор 118, и с одним или более мостами 120, которые соединены  
25 по высокоскоростным последовательным каналам или несколькими контроллерами 122-127, такими как контроллер 127, которые предоставляют доступ к всевозможным видам съемных накопителей 128, электронных дисплеев, устройств ввода и прочих подобных компонентов, подкомпонентов и вычислительных ресурсов.

На Фиг. 2A-D показаны два различных типа портативных устройств получения изображения. На Фиг. 2A-C показана цифровая камера 202. Цифровая камера содержит объектив 204, кнопку спуска затвора 205, нажатие которой пользователем приводит к получению цифрового изображения, которое соответствует отраженному свету, поступающему в объектив 204 цифровой камеры. С задней стороны цифровой камеры, которая видна пользователю, когда он держит камеру при съемке цифровых  
35 изображений, находится видоискатель 206 и жидкокристаллический дисплей видоискателя 208. С помощью видоискателя 206 пользователь может напрямую просматривать создаваемое объективом 204 камеры изображение, а с помощью жидкокристаллического дисплея 208 - просматривать электронное отображение создаваемого в настоящий момент объективом изображения. Обычно пользователь  
40 камеры настраивает фокус камеры с помощью кольца фокусировки 210, смотря при этом через видоискатель 206 или на жидкокристаллический экран видоискателя 208, чтобы выбрать требуемое изображение перед тем, как нажать на кнопку 205 спуска затвора для получения цифрового снимка изображения и его сохранения в электронной памяти цифровой камеры.

45 На Фиг. 2D показан типовой смартфон с передней стороны 220 и задней стороны 222. На задней стороне 222 имеется объектив 224 цифровой камеры и датчик приближения и (или) цифровой экспонометр 226. На передней стороне смартфона 220 под управлением приложения может отображаться получаемое изображение 226,

аналогично работе жидкокристаллического дисплея 208 видеоискателя цифровой камеры, а также сенсорная кнопка 228 спуска затвора, при прикосновении к которой происходит получение цифрового изображения и сохранение его в памяти смартфона.

На Фиг. 3 показано типовое изображение с цифровым кодированием. Кодированное изображение включает двумерный массив пикселей 302. На Фиг. 3 каждый небольшой квадрат, например, квадрат 304, является пикселем, который обычно определяется как наименьшая часть детализации изображения, для которой предусматривается цифровая кодировка. Каждый пиксель представляет собой место, обычно представленное парой цифровых значений, соответствующих значениям на осях прямоугольной системы координат  $x$  и  $y$ , 306 и 308 соответственно. Таким образом, например, пиксель 304 имеет координаты  $x, y$  (39,0), а пиксель 312 - координаты (0,0). Оси выбираются произвольно. Оси  $x$  и  $y$  могут быть взаимозаменены, например, в соответствии с другим правилом. В цифровой кодировке пиксель представлен числовыми значениями, указывающими на то, как область изображения, соответствующая пикселю, представляется при печати, отображается на экране компьютера или ином дисплее. Обычно для черно-белых изображений для представления каждого пикселя используется единичное значение в интервале от 0 до 255 с числовым значением, соответствующем уровню серого, на котором передается пиксель. Согласно общепринятому правилу значение «0» соответствует черному цвету, а значение «255» - белому. Для цветных изображений может применяться любое из множества различных числовых значений, указывающих на цвет. В одной из стандартных цветовых моделей, показанной на Фиг. 3, каждый пиксель связан с тремя значениями или координатами  $(r,g,b)$ , которые указывают на яркость красного, зеленого и синего компонента цвета, отображаемого в соответствующей пикселю области.

На Фиг. 4 показан один из вариантов цветовой модели RGB. Тремя координатами основных цветов  $(r,g,b)$  представлен весь спектр цветов, как было показано выше со ссылкой на Фиг. 3. Цветовая модель может считаться соответствующей точкам в пределах единичного куба 402, в котором трехмерное цветовое пространство определяется тремя осями координат: (1)  $r$  404; (2)  $g$  406; и (3)  $b$  408. Таким образом, координаты отдельного цвета находятся в интервале от 0 до 1 по каждой из трех цветковых осей. Например, чистый синий цвет максимально возможной яркости соответствует точке 410 по оси  $b$  с координатами (0,0,1). Белый цвет соответствует точке 412 с координатами (1,1,1), а черный цвет - точке 414, началу системы координат с координатами (0,0,0).

На Фиг. 5 показана другая цветовая модель, называемая цветовой моделью «оттенок-насыщенность-светлота» (HSL). В этой цветовой модели цвета содержатся в трехмерной бипирамидальной призме 500 с шестигранным сечением. Оттенок ( $h$ ) связан с доминантной длиной волны излучения света, воспринимаемого наблюдателем. Значение оттенка находится в интервале от  $0^\circ$  до  $360^\circ$ , начиная с красного цвета 502 в точке  $0^\circ$ , проходя через зеленый 504 в точке  $120^\circ$ , синий 506 в точке  $240^\circ$ , и заканчивая красным 502 в точке  $360^\circ$ . Насыщенность ( $s$ ), находящаяся в интервале от 0 до 1, обратно связана с количеством белого и черного цвета, смешанного при определенной длине волны или оттенке. Например, чистый красный цвет 502 является полностью насыщенным при насыщенности  $s=1,0$ , в то же время розовый цвет имеет насыщенность менее 1,0, но более 0,0, белый цвет 508 является полностью ненасыщенным с  $s=0,0$ , а черный цвет 510 также является полностью ненасыщенным с  $s=0,0$ . Полностью насыщенными цветами являются на периметре среднего шестигранника, содержащего точки 502, 504 и 506. Шкала оттенков серого проходит от черного 510 до белого 508 по центральной

вертикальной оси 512, представляющей полностью ненасыщенные цвета без оттенка, но с различными пропорциями черного и белого. Например, черный 510 содержит 100% черного и не содержит белого, белый 508 содержит 100% белого и не содержит черного, а исходная точка 513 содержит 50% черного и 50% белого. Светлота (l), представленная 5 центральной вертикальной осью 512, указывает на уровень освещенности в интервале от 0 для черного 510, при l=0,0, до 1 для белого 508, при l=1,0. Для произвольного цвета, представленного на Фиг. 5 точкой 514, оттенок определяется как угол  $\theta$  516, между 10 первым вектором из исходной точки 513 к точке 502 и вторым вектором из исходной точки 513 к точке 520, в которой вертикальная линия 522, проходящая через точку 514, пересекает плоскость 524, включающую исходную точку 513 и точки 502, 504 и 506. Насыщенность представлена отношением расстояния представленной точки 514 от 15 вертикальной оси 512 d' к длине горизонтальной линии, проходящей через точку 520 от исходной точки 513 к поверхности бипирамидальной призмы 500, d. Светлота представлена вертикальным расстоянием от контрольной точки 514 до вертикального уровня точки 510, представляющей черный цвет. Координаты конкретного цвета в цветовой модели HSL (h,s,l) могут быть получены на основе координат цвета в цветовой модели RGB (r,g,b) следующим образом:

$$l = \frac{(C_{\max} - C_{\min})}{2},$$

$$h = \begin{cases} 60^\circ \times \left( \frac{g-b}{\Delta} \bmod 6 \right), & \text{when } C_{\max} = r \\ 60^\circ \times \left( \frac{b-r}{\Delta} + 2 \right), & \text{when } C_{\max} = g \\ 60^\circ \times \left( \frac{r-g}{\Delta} + 4 \right), & \text{when } C_{\max} = b \end{cases}, \text{ и}$$

$$s = \begin{cases} 0, \Delta = 0 \\ \frac{\Delta}{1 - |2l - 1|}, & \text{otherwise} \end{cases},$$

где значения r, g и b соответствуют яркости красного, зеленого и синего первичных цветов, нормализованных на интервале [0, 1];  $C_{\max}$  представляет нормализованное 35 значение яркости, равное максимальному значению из r, g и b;  $C_{\min}$  представляет собой нормализованное значение яркости, равное минимальному значению из r, g и b; а  $\Delta$  определяется как  $C_{\max} - C_{\min}$ .

На Фиг. 6 изображено формирование полутонового или бинарного изображения из 40 цветного изображения. В цветном изображении каждый пиксель обычно связан с тремя значениями: a, b и c 602. В разных цветовых моделях для представления конкретного цвета используются разные значения a, b и c. Полутоновое изображение содержит для каждого пикселя только одно значение яркости 604. Бинарное изображение является частным случаем полутонового изображения, которое имеет только два значения 45 яркости «0» и «1». Обычно полутоновые изображения могут иметь 256 или 65 536 разных значений яркости для каждого пикселя, представленного байтом или 16-битным словом соответственно. Таким образом, чтобы преобразовать цветное изображение в полутоновое, три значения a, b и c в каждом цветном пикселе необходимо преобразовать в одно значение яркости для соответствующего пикселя полутонового или бинарного

изображения. На первом этапе три значения цвета a, b и c, преобразуются в значение яркости L, обычно в интервале [0,0, 1,0] 606. Для определенных цветовых моделей к каждому из цветовых значений 608 применяется нетривиальная функция, и результаты суммируются 610, давая значение яркости. В других цветовых моделях каждое цветовое значение умножается на коэффициент, и полученные результаты суммируются 612, давая значение яркости. В некоторых цветовых системах одно из трех цветовых значений является, фактически, значением 614 яркости. Наконец, в общем случае к трем цветовым значениям 616 применяется функция, которая дает значение яркости. Затем значение яркости квантуется 618, позволяя получить значение яркости оттенков серого в требуемом интервале, обычно [0, 255] для полутоновых изображений и одно из двух значений яркости (0,1) для бинарных изображений.

На Фиг. 7 показано дискретное вычисление градиента яркости. На Фиг. 7 показан небольшой квадратный участок 702 цифрового изображения. Каждая клетка, например, клетка 704, представляет пиксель, а числовое значение в клетке, например, значение «106» в клетке 704, представляет яркость серого цвета. Допустим, пиксель 706 имеет значение яркости «203». Этот пиксель и четыре смежных с ним пикселя показаны на крестообразной схеме 708 справа от участка 702 цифрового изображения. Рассматривая левый 710 и правый 712 соседние пиксели, изменение значения яркости в направлении x,  $\Delta x$  можно дискретно вычислить как:

$$\Delta x = \frac{247 - 150}{2} = 48.5.$$

Рассматривая нижний 714 и верхний 716 соседние пиксели, изменение значения яркости в вертикальном направлении  $\Delta y$  можно вычислить как:

$$\Delta y = \frac{220 - 180}{2} = 20.$$

Вычисленное значение  $\Delta x$  является оценкой частного дифференциала непрерывной функции яркости относительно оси x в центральном пикселе 706:

$$\frac{\partial F}{\partial x} \cong \Delta x = 48.5.$$

Частный дифференциал функции F яркости относительно координаты y в центральном пикселе 706 рассчитывается по  $\Delta y$ :

$$\frac{\partial F}{\partial y} \cong \Delta y = 20.$$

Затем градиент яркости в пикселе 706 может быть рассчитан следующим образом:

$$\text{gradient} = \nabla F = \frac{\partial F}{\partial x} \mathbf{i} + \frac{\partial F}{\partial y} \mathbf{j} = 48.5 \mathbf{i} + 20 \mathbf{j}$$

где  $\mathbf{i}$  и  $\mathbf{j}$  представляют собой единичные векторы в направлениях x и y. Модуль вектора градиента и угол вектора градиента далее рассчитываются следующим образом:

$$|\text{gradient}| = \sqrt{48.5^2 + 20^2} = 52.5$$

$$\theta = \text{atan2}(20, 48.5) = 22.4^\circ$$

Направление вектора 720 градиента яркости и угол  $\theta$  722 показаны наложенными на участок 702 цифрового изображения на Фиг. 7. Следует учесть, что вектор градиента направлен в сторону наиболее резкого увеличения яркости от пикселя 706. Модуль вектора градиента указывает на ожидаемое увеличение яркости на единицу приращения

в направлении градиента. Конечно же, поскольку градиент оценивается исключительно с помощью дискретных операций, в вычислении, показанном на Фиг. 7, направление и модуль градиента представлены исключительно оценками.

На Фиг. 8 показан градиент, рассчитанный для некоей точки на непрерывной поверхности. На Фиг. 8 представлена непрерывная поверхность  $z=F(x,y)$ . Непрерывная поверхность 802 строится относительно трехмерной декартовой системы координат 804 и имеет похожую на шляпу форму. Для отображения непрерывного множества точек с постоянным значением  $z$  на поверхности могут быть построены контурные линии, например, контурная линия 806. В конкретной точке 808 на контуре, построенном на поверхности, вектор градиента 810, рассчитанный для точки, расположен перпендикулярно к контурной линии и точкам в направлении максимально резкого наклона вверх на поверхности от точки 808.

Обычно вектор градиента яркости расположен перпендикулярно к границе яркости, причем чем больше модуль градиента, тем данная граница более четкая. Граница имеет различия в яркости пикселей на любой из сторон границы с общими признаками, или, другими словами, яркость в соседних строках пикселей, перпендикулярных границе, меняется сходным образом. На Фиг. 9 показан ряд примеров градиента яркости. Каждый пример, такой как пример 902, содержит центральный пиксель, для которого рассчитывается градиент, и четыре прилегающих пикселя, которые используются для расчета  $\Delta x$  и  $\Delta y$ . Границы с наибольшей яркостью показаны в первой колонке 904. В этих случаях модуль градиента составляет не менее 127,5, а в третьем случае 906 - 180,3. При относительно небольшой разности по границе, показанной в примере 908, получается градиент величиной всего 3,9. Во всех случаях вектор градиента расположен перпендикулярно очевидному направлению границы яркости, проходящей через центральный пиксель.

Многие методы обработки изображений включают применение ядер к сетке пикселей, составляющей изображение. На Фиг. 10 показано применение ядра к изображению. На Фиг. 10 небольшая часть изображения I 1002 представлена в виде прямоугольной сетки пикселей. Ниже изображения I 1002 показано небольшое ядро  $3 \times 3$  k 1004. Для выполнения в отношении изображения операции обработки на основе ядра ядро обычно применяется к каждому пикселю изображения. В случае ядра  $3 \times 3$ , такого как ядро k 1004, показанное на Фиг. 10, для пикселей на границе можно использовать модифицированное ядро, также изображение можно раздвинуть, скопировав значения яркости для пикселей границы в описывающий прямоугольник из пикселей, чтобы иметь возможность применять ядро к каждому пикселю исходного изображения. Чтобы применить ядро к пикселю изображения, ядро 1004 численно накладывается на окрестность пикселя 1006 на изображении с такими же размерами в пикселях, что и у ядра. Применение ядра к окрестности пикселя, к которому применяется ядро, позволяет получить новое значение для пикселя в преобразованном изображении, полученном при выполнении операции на основе ядра. Для некоторых типов ядер и операций на основе ядра новое значение пикселя, к которому применяется ядро,  $I_n$ , вычисляется как сумма произведений значения ядра и пикселя, соответствующего значению 1008 ядра. В других случаях новое значение пикселя является более сложной функцией окрестности для пикселя и ядра 1010. В некоторых других типах обработки изображений новое значение пикселя генерируется функцией, применяемой к окрестностям пикселя без использования ядра 1012.

На Фиг. 11 показана свертка ядра с изображением с получением преобразованного изображения. Как правило, ядро последовательно применяется к каждому пикселю

изображения. В некоторых случаях ядро применяется только к каждому пикселю изображения, не принадлежащему границе изображения. На Фиг. 11 ядро  $3 \times 3$ , выделенное штриховкой 1102, было последовательно применено к первой строке пикселей, не принадлежащих границе изображения 1104. Каждое новое значение, созданное в результате применения ядра к пикселю в исходном изображении 1106, было перенесено в преобразованное изображение 1107. Другими словами, ядро последовательно применяется к исходным окрестностям каждого пикселя на исходном изображении для получения преобразованных значений пикселей, включенных в преобразованное изображение. Данный процесс называется «сверткой» и отчасти связан с математической операцией свертки, которая выполняется путем умножения изображений, к которым применено преобразование Фурье, с последующим обратным преобразованием Фурье по произведению.

На Фиг. 12 показан пример ядра и методик обработки изображений на основе ядра. В процессе, называемом «медианной фильтрацией», значения яркости некоего пикселя и пикселей в окрестности данного пикселя в исходном изображении 1202 сортируются 1204 по возрастанию величины, и медианное значение 1206 выбирается в качестве значения 1208 для данного пикселя в преобразованном изображении. Гауссово сглаживание и очистка от шумов включают применение гауссова ядра 1210 ко всем окрестностям 1214 исходного изображения для создания значения для центрального пикселя окрестности 1216 в соответствующей окрестности обработанного изображения. Значения в гауссовом ядре рассчитываются по выражению, например, по выражению 1218 для создания дискретного представления гауссовой поверхности над окрестностью, образованного вращением кривой нормального распределения вокруг вертикальной оси, совпадающей с центральным пикселем. Горизонтальные и вертикальные компоненты градиента изображения для каждого пикселя оцениваются применением соответствующих ядер градиента  $G_x$  1220 и  $G_y$  1222. Были указаны только три из множества различных типов методик обработки изображения на основе свертки.

Способы и системы, к которым относится настоящий документ

На Фиг. 13А-В показан один вариант реализации способа, к которому относится настоящий документ. Данный вариант реализации специально направлен на исправление искривления текстовых строк на разворотах книг. Тем не менее, тот же способ либо аналогичные ему способы могут использоваться для исправления широкого диапазона искривлений текстовых строк, которые встречаются во многих видах содержащих текст цифровых изображений.

На Фиг. 13А показан разворот книги. Разворот книги 1302 содержит две страницы 1304 и 1306. Так как данные страницы соединены по корешку книги 1308, то при ее открытии и помещении на поверхность обложкой вниз поверхность страниц принимает форму дуги, направленной вверх и от корешка. Таким образом, на изображении разворота книги, полученном с помощью цифровой камеры, текстовые строки, такие как текстовая строка 1310, являются изогнутыми не только потому, что фактически изогнуты поверхности страниц, но и ввиду искривления, вызванного искажением перспективы. Тем не менее, если разгладить страницу и смотреть на нее сверху вниз, текстовые строки будут линейными, а символы и слова, содержащиеся в тексте, будут расположены в знакомой форме прямоугольника. При изгибании и искривлении текстовых строк из-за искажения перспективы и изгиба страниц раскрытой книги буквы и символы в текстовых строках искажаются и нелинейно сжимаются, что существенно усложняет задачу автоматизированного оптического распознавания символов и букв на содержащем текст изображении.

На Фиг. 13В показаны этапы, выполняемые в рамках способа настоящего изобретения при обработке изображения раскрытой книги, как описано выше со ссылкой на Фиг. 13А. На первом этапе изображение разворота книги 1320 обрабатывается с целью обособить только ту часть изображения, которая соответствует двум страницам данной книги 1322. На следующем этапе для каждой из двух страниц 1324 и 1326 подготавливаются отдельные изображения. На каждом из этих двух изображений одной страницы страница, если она не ориентирована вертикально на исходном изображении 1320, вращается до достижения вертикальной ориентации. На следующем этапе, выполняемом в отношении каждого из двух изображений 1324 и 1326, изображение одной страницы обрабатывается таким образом, чтобы граница страницы, не прилегающая к корешку, была также ориентирована вертикально либо параллельно к прилегающей к корешку границе, с получением изображений 1328 и 1330. На окончательном этапе изогнутые верхняя и нижняя границы каждого изображения выпрямляются вместе с текстовыми строками на странице с получением окончательных обработанных изображений 1332 и 1334, на каждом из которых страница представлена в виде прямоугольника, так словно ее изображение было получено из точки непосредственно над центроидом страницы при расположении оптической оси устройства получения изображения перпендикулярно к данной странице. Существует множество дополнительных операций, включая изменение масштаба, повышение контурной резкости и прочие операции обработки изображений, которые могут быть выполнены для дальнейшего улучшения изображения страницы для оптического распознавания символов.

На Фиг. 14А-С показана первоначальная обработка изображения страницы для подготовки к выравниванию границ страницы параллельно корешку, как описано выше со ссылкой на Фиг. 13В. Как показано на Фиг. 14А, исходное изображение страницы 1402 свертывается с помощью Гауссова ядра, либо в отношении изображения применяется метод медианной фильтрации с получением сглаженного изображения 1404 без шумов. Затем к сглаженному изображению без шумов применяются два ядра градиента  $G_x$  и  $G_y$  с получением двух карт компонентов градиента, из которых формируется карта градиентов 1406, как описано выше со ссылкой на Фиг. 7-9. На карте градиента каждый пиксель, такой как пиксель 1408, соотносится с направлением градиента и величиной градиента. По карте градиентов могут быть определены границы, такие как контур страницы на изображении страницы, представленные в виде строки или кривой пикселей с соотнесенными с ними векторами градиента, имеющими сходную ориентацию, и направление, перпендикулярное данной строке или кривой, и относительно большую величину. Например, как показано на Фиг. 14А, вертикальная граница 1410 на изображении будет соответствовать колонке 1412 пикселей на карте градиентов с векторами большой величины, указывающими в одном и том же направлении и перпендикулярными данной границе. Для извлечения границ из карты градиентов, соответствующих границам яркости на исходном изображении, может быть применена пороговая фильтрация величин градиента и различные методики на основе непрерывности. Другие хорошо известные способы, предназначенные для определения границ на изображении страницы, применяются (в альтернативных вариантах реализации) аналогичным образом.

Как показано на Фиг. 14В, для утончения границ, так чтобы из изображения страницы можно было извлечь четко очерченные границы, используется методика подавления немаксимальных величин, применяемая как функция к окрестностям пикселей на карте градиентов. На Фиг. 14В для принятия решения о том, нужно ли сохранять значение

центральной ячейки или элемента 1422 преобразованного изображения 1424, используется окрестность  $3 \times 3$  1420 вокруг центральной ячейки или элемента 1422 карты градиентов. Если центральный пиксель имеет максимальную величину по направлению градиента в данной окрестности 1426, то значение сохраняется 1428 в преобразованной карте градиентов, полученной путем подавления немаксимальных величин. Таким образом, немаксимальные величины градиента, перпендикулярные границе, подавляются или устанавливаются на 0, с тем чтобы утончить многопиксельную границу 1430 в исходной карте градиентов до однопиксельной границы 1432 в преобразованной карте градиентов.

10 Как показано на Фиг. 14С, к карте градиентов 1406 применяются подавление немаксимальных величин и фильтрация длин границ с получением карты границ 1436, из которой, используя границы из нее, а также дополнительные геометрические характеристики и характеристики яркости страниц, может быть извлечен контур страницы 1440. Дополнительные характеристики могут включать информацию о цветах из исходного изображения разворота книги, используемую для дифференциации текстовых пикселей от фоновых пикселей, ожидаемых формах и размерах книг и страниц книг, а также другую информацию, используемую для фильтрации границ страниц от других дополнительных границ яркости, которые встречаются при операции обработки для определения границ, описанной выше со ссылкой на Фиг. 14А-В.

20 На Фиг. 15А-В показан один из подходов к процессам выравнивания вертикальной границы страницы или исправления вертикальной перспективы страницы, показанной выше со ссылкой на Фиг. 13В. Как показано на Фиг. 15А, контур страницы, извлеченный из карты градиентов с подавленными немаксимальными величинами, накладывается на исходное изображение страницы для формирования страницы 1502 на изображении страницы 1504. Затем изображение страницы вращается, с тем чтобы прилегающая к корешку граница 1506 страницы была вертикальной. На данном этапе процесса верхняя 1508 и нижняя 1512 границы страницы обычно являются изогнутыми, граница страницы, параллельная корешку 1510, но не прилегающая к нему, может быть отклонена от вертикального направления на угол  $\alpha$  1511, а нижняя граница страницы 1512 может быть расположена с уклоном по отношению к границе 1510 на угол  $\beta$  1513 и отклонена от вертикального направления на угол  $\gamma$  1514. На первом этапе для корректировки положения неприлегающей к корешку границы до вертикальной ориентации осуществляется смещение пикселей на изображении, как показано на промежуточном изображении 1516, а на втором этапе осуществляется реорганизация пикселей на изображении для корректировки положения нижней границы 1512 до горизонтального 1518.

На Фиг. 15В показан один из способов корректировки ориентации границы страницы. На Фиг. 15В показан процесс корректировки положения границы, параллельной корешку, но не прилегающей к нему, так чтобы она имела вертикальную ориентацию. 40 Проводится корректировка каждой горизонтальной строки пикселей на изображении. Для горизонтальной строки пикселей 1520 на Фиг. 15В приведен ряд значений, полученных на основе геометрических характеристик границы 1510 на содержащем текст изображении. Верхний угол страницы 1522 имеет координаты  $(x, y)$ . Пересечение 1524 горизонтальной строки пикселей 1520 и границы 1510 имеет координаты  $(x', y')$ . 45 Ширина страницы имеет значение  $X$  1526. Точка 1524 делит горизонтальную строку пикселей на два фрагмента, один длиной  $x'$  1527, а другой длиной  $X - x'$  1528. Чтобы исправить положение горизонтальных строк пикселей 1520 для приведения границы 1510 в вертикальное положение, необходимо удалить пиксели  $x - x'$  1530 из второй части

1528 горизонтальной строки и добавить то же самое число пикселей в первую часть  
 1527 горизонтальной строки пикселей. Горизонтальная строка пикселей 1532 снова  
 показана в правой верхней части Фиг. 15В. В скопированном изображении 1534  
 небольшие вертикальные стрелки, помеченные буквой «d», такие как стрелка 1536,  
 5 указывают на пиксели, которые дублируются в первой части строки пикселей, а  
 небольшие вертикальные стрелки, помеченные буквой «r», такие как стрелка 1538,  
 указывают на пиксели, подлежащие удалению из второй части строки пикселей. Более  
 длинные стрелки, такие как стрелка 1540, иллюстрируют преобразование пикселей в  
 исходной горизонтальной строке пикселей 1532 в пиксели в преобразованной  
 10 горизонтальной строке пикселей 1542 после совершения операций дублирования  
 пикселей и удаления пикселей. Как можно видеть, сравнив исходную горизонтальную  
 строку пикселей 1534 с преобразованной горизонтальной строкой пикселей 1542,  
 значение яркости пикселей в положении  $x'$  в исходной строке пикселей, обозначенное  
 заштрихованной областью 1544, сместилось в преобразованной горизонтальной строке  
 15 пикселей 1542 вправо 1546. Значения яркости пикселей, добавленных к первой части  
 строки пикселей, и значения яркости двух пикселей, прилегающих к каждому пикселю,  
 удаленному из второй части строки пикселей, могут быть изменены во избежание резких  
 перепадов в градиентах яркости пикселя. Как показано на Фиг. 15В, во избежание  
 локальных искажений пиксели дублируются и удаляются в строке пикселей через  
 20 фиксированные интервалы. В других вариантах реализации для корректировки  
 ориентации границы страницы на изображении страницы или для исправления  
 вертикальной перспективы страницы могут использоваться многие другие методики и  
 разновидности методики, проиллюстрированной на Фиг. 15В. Так, в некоторых случаях  
 граница страницы может быть не видна на исходном изображении, поэтому для  
 25 выполнения эквивалентной операции без использования данной границы страницы  
 могут применяться другие указатели или ориентиры на изображении. Показанный  
 способ немного изменяется при корректировке границы, параллельной корешку, но  
 не прилегающей к нему, на изображении разворота книги, состоящего из двух страниц.

На Фиг. 16А-Г показано разделение фрагментов на блоки. Как показано на Фиг.  
 30 16А, строка текста 1602 на изображении страницы может быть изогнута. На первом  
 этапе разделения фрагментов на блоки, как показано на Фиг. 16В, осуществляется  
 распознавание слов, фрагментов слов и (или) других скоплений смежных символов и  
 знаков в строке текста с применением подхода, не основанного на методиках  
 оптического распознавания символов, требующих большого объема вычислений. В  
 35 определенных вариантах реализации распознавание фрагментов осуществляется на  
 бинарной версии изображения страницы, при этом пикселям, содержащимся в знаках  
 и символах (называемым символьными пикселями), присваивается одно из двух значений,  
 а несимвольным пикселям присваивается второе из данных двух значений. Как описано  
 ниже, различие между символьными и несимвольными пикселями может быть  
 40 установлено с помощью порога яркости либо более сложных процессов. На Фиг. 16В  
 каждый из выявленных фрагментов и фрагментов нескольких слов показан в  
 прямоугольнике, таком как прямоугольник 1604. В нижеследующем описании термин  
 «фрагмент» означает скопления символов, включающие слова, фрагменты слов и  
 скопления фрагментов нескольких слов. Один из возможных подходов к определению  
 45 слов, фрагментов слов и других скоплений символов и знаков на изображении страницы,  
 не основанный на методе оптического распознавания символов, показан на Фиг. 16С-  
 Г. На первом этапе, как показано на Фиг. 16С, по изображению страницы  
 распределяется некое число отправных, или затравочных, точек. На Фиг. 16С начальная

отправная точка 1606 расположена над конечным знаком слова. Из начальной отправной точки осуществляется поиск дискообразного пространства радиусом  $r$  1607, с тем чтобы найти все символьные пиксели на дискообразной области. Символьные пиксели могут быть определены как пиксели со значением яркости ниже порогового -

5 в случае содержащих текст изображений, на которых символы являются черными на белом или светлом фоне. На Фиг. 16D две небольших области 1608 и 1609 конечного знака в слове, показанные затененными на Фиг. 16D, включают символьные пиксели на дискообразной области и формируют ядро слова или фрагмента слова. Как показано на Фиг. 16E, от символьных пикселей, определенных в предыдущей итерации поиска

10 1612, осуществляются аналогичные операции поиска, и процесс продолжается, как показано на Фиг. 16F. Наконец, как показано на Фиг. 16G, в рамках радиуса  $r$  от любого из пикселей, определенных для слова или фрагмента, не может быть найдено ни одного дополнительного символьного пикселя, в результате чего поиск заканчивается и фрагмент, заключенный в очерченный пунктирной линией прямоугольник 1614,

15 становится распознанным словом или фрагментом слова. Следует отметить, что для распознавания именно слов и фрагментов слов, а не скоплений слов, радиус  $r$  выбирается таким образом, чтобы он был значительно короче среднего или обычного расстояния  $s$  1616 между словами. Тем не менее, во многих случаях поиск фрагментов, не основанный на распознавании символов, может завершиться с возвратом фрагментов,

20 которые содержат части двух или более слов, а также скопления слов.

Далее осуществляется разделение на блоки первоначально распознанных фрагментов. На Фиг. 17A-D показан процесс разделения фрагментов на блоки. Как показано на Фиг. 17A, координаты 1702 для центроида 1704 некоего фрагмента легко вычисляются как средняя  $x$  координата 1706 и средняя  $y$  координата 1708 для всех символьных

25 пикселей  $i$  во фрагменте. Затем, как показано на Фиг. 17B, угол  $\alpha$  1710 наклона основной оси 1712 фрагмента по отношению к горизонтальному направлению 1713 может быть вычислен как задача минимизации 1714, в которой угол  $\alpha$  выбирается как угол, который минимизирует расстояния  $d$  1716 и 1717 всех символьных пикселей, включая символьные пиксели 1718 и 1719, во фрагменте.

30 На Фиг. 17C показан один из способов определения угла наклона  $\alpha$ . Сначала координаты символьного пикселя представляются в табличной форме в виде таблицы 1720. Далее координаты преобразуются в координаты, соотносительные с рассчитанным центроидом с координатами  $(X, Y)$  и представляются в табличной форме в виде таблицы 1722. Из этих преобразованных координат вычисляется три значения  $K_1$ ,  $K_2$  и  $K$  1724-

35 1726. Тангенс  $a$  дается в виде выражения 1728. Есть два возможных угла, данные в виде выражения 1728, которые соответствуют двум ортогональным основным осям фрагмента слова. Углом наклона  $\alpha$  выбирается тот угол из двух вычисленных углов, который находится ближе всего к горизонтальному углу 0 1730. В соответствии с альтернативным вариантом может быть выбран угол, представляющий угол

40 центральной оси фрагмента по отношению к горизонтальному направлению. С правой стороны Фиг. 17C 1732 приводится вывод для данного способа. Сначала при вращении фрагмента на угол  $\alpha$  координаты пикселей преобразуются в соответствии с умножением матрицы вращения на вектор исходных координат 1734. Значение  $N$  вычисляется как сумма координат  $u$  в квадрате для символьных пикселей 1736, выраженных в виде

45 преобразованных координат 1722. Выражение для координаты  $u^{0*}$  пикселя после вращения пикселя на угол  $\alpha$  вокруг центроида вычисляется из выражения матрицы как выражение 1738. Производная данного выражения по отношению к углу  $\alpha$  представляется выражениями 1740 и 1742. Затем производная значения  $N$  по отношению

к углу вращения  $\alpha$  представляется выражением 1744. Сумма расстояний  $d_i$  от пикселей до основной оси является минимизированной, когда производное от  $H$  по отношению к  $\alpha$  равно 0, как показано выражением 1746. Решение данного выражения 1748 приводит к квадратичной формуле в  $\alpha$  1750. Решением данной квадратичной формулы является выражение 1728. После определения центроида и угла наклона  $\alpha$  для некоего фрагмента данный фрагмент заключается в минимальном вмещающем прямоугольнике 1760, как показано на Фиг. 17D. В альтернативных вариантах реализации вместо прямоугольников могут использоваться параллелограммы или четырехугольники. Углом наклона главной оси минимального вмещающего прямоугольника по отношению к горизонтали обычно является вычисленный угол  $\alpha$ . В случаях, когда угол  $\alpha$  выбран неверно как угол для другой основной оси, последующая фильтрация фрагментов приводит к удалению фрагментов с неверно выбранными углами наклона.

На Фиг. 18A-B показан первый этап фильтрации, применяемый к начальным блокам фрагментов, полученным с помощью способа разделения фрагментов на блоки, описанного со ссылкой на Фиг. 17A-D. Как схематически показано на Фиг. 18A, фильтр блоков фрагментов применяется к начальным блокам фрагментов на изображении страницы 1802 для удаления нежелательных блоков фрагментов с получением отфильтрованных блоков фрагментов 1804. На Фиг. 18B показан один из фильтров, которые могут быть применены к блокам фрагментов. Каждый блок фрагментов имеет угол наклона по отношению к горизонтали  $\alpha$  1810, длину длинной стороны  $b$  1812 и длину короткой стороны  $a$  1814. Проекция стороны 1814 длины  $a$  на горизонтальное направление имеет длину  $c$ . Блок фрагмента 1816 отклоняется, или отфильтровывается, когда отношение  $a$  к  $c$  меньше первого порога  $T_1$ , когда длина  $a$  меньше второго порога  $T_2$  и (или) когда отношение  $a$  к высоте изображения или отношение  $b$  к ширине изображения больше третьего порога  $T_3$  1818. Первый критерий фильтрации эквивалентен отклонению блока фрагмента, когда  $\alpha$  больше порогового угла. Для отбора наиболее подходящих блоков фрагментов с формированием начальной модели искривления изображения страницы могут использоваться другие критерии фильтрации, включая требование, чтобы длины  $a$  и  $b$  не выходили за рамки определенных диапазонов, требование, чтобы площадь минимального вмещающего прямоугольника не выходила за рамки некоего диапазона, и прочие критерии.

На Фиг. 19A-B показано формирование исходной модели Искривления текстовых строк на изображении страницы. Каждому пикселю на изображении страницы 1908, который не выходит за границы блока фрагмента, присваивается угол наклона, равный углу наклона  $\alpha$  блока фрагмента, в котором расположены данные пиксели. На Фиг. 19A изображен процесс присвоения угла наклона  $\alpha$  1902 пикселю 1904 изображения страницы на блоке фрагмента 1906 изображения страницы 1908. Углы наклона присваиваются только пикселям, совпадающим с центроидами блоков фрагментов. Затем к координатам и присвоенным углам наклона  $\alpha$  применяется способ согласования данных 1910 по всем пикселям в центроидах блоков фрагментов для определения функции  $f(x,y)$  1912, которая возвращает тангенс угла наклона  $\alpha$  в координатах  $x, y$  на изображении страницы 1908. В одном из вариантов реализации функция  $f(x,y)$  является полиномом со степенью 3 для  $x$  и степенью 1 для  $y$  1914 с восемью коэффициентами  $a_1$ - $a_8$ . Значения данным коэффициентам присваиваются в рамках способа согласования данных 1910. Следует отметить, что функция  $f(x,y)$ , называемая «исходной моделью», возвращает угол наклона  $\alpha$  для любого положения  $(x,y)$  на изображении страницы, а не только для положения в блоках фрагментов. Другими словами, исходная модель

имеет общий характер, но формируется путем согласования данных из подмножества пикселей на изображении.

На Фиг. 19В показан один из способов, именуемый «метод Гаусса-Ньютона» согласования угла наклона и данных о координатах, полученных из блоков фрагментов, с полиномиальной функцией  $f(x,y)$ . Сначала на Фиг. 19В заново формулируется задача согласования данных. Коэффициенты  $\beta_1, \beta_2, \dots, \beta_8$  могут рассматриваться как вектор  $\beta$  1922. Для значений данных коэффициентов строится некое предположение с формированием исходного вектора  $\beta$ . Разница между тангенсами, возвращенными функцией  $f(x,y)$  с текущими значениями  $\beta$  и тангенсами измеренных углов наклона, выражается как множество  $n$  невязок  $r_i$  1924, где  $n$  - это число пикселей с присвоенными углами наклона  $\alpha$ . Каждая невязка означает разницу между тангенсом угла наклона  $\alpha$ , присвоенного пикселю, и угла наклона, вычисленного для данного пикселя функцией  $f(x,y)$  с предположением текущих значений коэффициентов  $\beta_c$ . Новое значение вектора коэффициентов  $\beta_n$  может быть вычислено из текущих значений вектора коэффициентов  $\beta_c$ , из вычисленных невязок и определителя матрицы Якоби  $J_f$  для функции  $f(x,y)$  1926, как показано в выражении 1928. Следует отметить, что метод Гаусса-Ньютона минимизирует сумму возведенных в квадрат невязок 1930. Метод Гаусса-Ньютона иллюстрируется небольшой блок-схемой 1932. Выполнение данного способа начинается с исходного  $\beta$ , функции  $f(x,y)$  и множества данных 1934. В итерационном цикле при выполнении данного способа из множества данных и функции  $f(x,y)$ , делающей предположение о текущих коэффициентах  $\beta_c$ , на этапе 1936 вычисляются невязки  $r_1, r_2, \dots, r_n$ , где  $r$  - это вектор невязок. Затем с использованием выражения 1928 на этапе 1938 вычисляется новое множество коэффициентов  $\beta_n$ . Если разница между новыми и предыдущими коэффициентами меньше порогового значения, которое определяется на этапе 1940, новые коэффициенты возвращаются как коэффициенты для функции  $f(x,y)$  1942. В противном случае на этапе 1944  $\beta_c$  присваивается  $\beta_n$  и выполняется другая итерация. Данная блок-схема является в некоторой степени упрощенной, так как метод Гаусса-Ньютона не гарантированно обеспечивает сходимость метода. Таким образом, помимо проверки на разность, выполняемой на этапе 1940, данный метод обычно применяет отсечку числа итераций и другие методики, позволяющие избежать бесконечной итерации метода. В альтернативных вариантах реализации могут использоваться другие способы согласования данных, такие как нелинейный метод сопряженных градиентов, разновидности многих других методик согласования искривлений, включая нелинейные способы согласования искривлений, такие как метод Левенберга-Марквардта. В одном из вариантов реализации на основе возведенных в квадрат значений углов наклона для каждого центра блока фрагментов путем формирования производных параметров получается система линейных уравнений, которая решается с помощью метода Гаусса, а не с помощью итеративного способа согласования данных.

После формирования исходной модели способом согласования данных, таким как способ, описанный выше со ссылкой на Фиг. 19А-В, снова осуществляется фильтрация блоков фрагментов. На Фиг. 20 приведена блок-схема вторичной фильтрации блоков фрагментов. На этапе 2002 фильтр получает модель  $f(x,y)$ , изображение страницы, текущее множество блоков фрагментов  $fSet$  и пустое множество результатов  $nSet$ . На этапе 2004 локальная переменная `num_removed` устанавливается на 0. В цикле `for` этапов 2006-2015 рассматривается каждый блок из множества блоков фрагментов во множестве

*fSet*. Локальной переменной *disparity* (несоответствие) на этапе 2007 задается значение 0. Затем во внутреннем цикле *for* этапов 2008-2010 вычисляется сумма невязок для пикселей в рассматриваемом в текущий момент блоке фрагмента. На этапе 2011 вычисляется среднее значение несоответствия, и это значение присваивается локальной переменной *disparity*. Если среднее несоответствие больше порогового значения, что определяется на этапе 2012, то на этапе 2013 локальная переменная *num\_removed* увеличивается. В противном случае блок фрагмента перемещается во множество блоков фрагментов *nSet*. Если остались другие подлежащие оценке блоки фрагментов, что определяется на этапе 2015, процесс управления возвращается к этапу 2007. В противном случае на этапе 2016 фильтр определяет, превышает ли число удаленных блоков фрагментов пороговое значение. Если число удаленных блоков фрагментов превышает пороговое значение, модель вычисляется заново из текущего множества блоков фрагментов в переменной-множестве *nSet*, в результате чего выполняется возврат *nSet* и повторно вычисленной модели. В противном случае возвращается исходное множество блоков фрагментов *fSet* в полученной модели.

На Фиг. 21А-Е показаны операции дилатации, эрозии, замыкания и открытия, используемые в описываемом варианте реализации способов настоящего изобретения для формирования закрашенных контуров, соответствующих текстовым строкам на содержащем текст изображении. На всех Фиг. 21А-Е используются одни и те же правила иллюстрации, далее описанные применительно к Фиг. 21А. На левой стороне Фиг. 21А показана небольшая часть содержащего текст изображения 2102. Это бинарное изображение, в котором значение «0» обозначает черный или текстовый пиксель, а значение «1» - пустое пространство. Путем свертки изображения 2102 с горизонтальным ядром 2104 выполняется операция горизонтальной дилатации 2103. Если центральный пиксель ядра перекрывает пиксель на изображении, пиксель на изображении, полученный в результате свертки, получает значение минимального значения пикселей на изображении, перекрытого пикселями ядра. Это ведет к эффекту дилатации, или горизонтального расширения, содержащих темные пиксели областей изображения. Результатом применения трехпиксельного ядра горизонтальной дилатации 2104 к изображению 2102 является дилатированное изображение 2105. На Фиг. 21В показана операция горизонтальной эрозии. Операция горизонтальной эрозии в некотором смысле имеет действие, противоположное действию операции горизонтальной дилатации. В операции горизонтальной эрозии трехпиксельное горизонтальное ядро 2110 свертывается с исходным изображением 2102, при этом пиксель изображения, перекрытый центральным пикселем в ядре 2110, получает на изображении, полученном в результате свертки, максимальное значение из всех пикселей на изображении, перекрытых пикселями в ядре. Это ведет к эффекту эрозии, или утончению, всех содержащих черные пиксели областей на изображении. На Фиг. 21В применение ядра эрозии через процесс свертки к исходному изображению 2102 ведет к формированию эродированного изображения 2112. Следует отметить, что примеры, приведенные на Фиг. 21А-Е, являются искусственными, так как характеристики символов на страницах текста обычно имеют ширину более одного пикселя.

Как показано на Фиг. 21С, в случае эрозии изначально дилатированного изображения 2105, полученного в результате свертки исходного изображения с оператором горизонтальной дилатации, путем свертки дилатированного изображения с оператором горизонтальной эрозии, получается замкнутое изображение 2114. Операция замыкания, таким образом, - это двухэтапная операция, при которой к изображению сначала применяется операция дилатации, а затем эрозии.

На Фиг. 21D-E показана операция открытия. Операция открытия противоположна по действию операции замыкания и заключается в применении операции эрозии к изображению с последующим применением операции дилатации. На Фиг. 21D замкнутое изображение 2114, полученное путем дилатации и эрозии изображения, как описано выше со ссылкой на Фиг. 21C, подвергается эрозии с получением промежуточного изображения 2116. На Фиг. 21E к промежуточному изображению 2116 применяется операция дилатации с получением открытого изображения 2118. Общее описание методики, применяемой в описываемом варианте реализации способа настоящего изобретения, приводится на Фиг. 21F, на котором к исходному изображению 2102 применяется операция дилатации, затем эрозии, снова эрозии и затем дилатации, то есть сначала замыкается, а затем открывается с получением замкнутого и затем открытого изображения 2118. Как понятно из Фиг. 21F, операция замыкания с последующим открытием заменяет знаки в строке текста на закрашенный контур строки текста 2120.

На Фиг. 22 показана операция замыкания с последующим открытием, выполняемая в отношении содержащего текст изображения для получения закрашенных контуров, соответствующих строкам текста. Содержащее текст изображение 2202 подвергается замыканию 2204, затем открытию 2206 с последующей операцией вертикального сглаживания 2208, при которой открытое изображение свертывается с гауссовым ядром в виде столбца, либо другим ядром, что имеет тот же эффект, для разглаживания изображения в вертикальном направлении. Результат 2210 - это изображение, в котором текстовые строки заменяются на горизонтальные затемненные прямоугольники или закрашенные контуры, соответствующие текстовым строкам. Как правило, в одном из вариантов реализации горизонтальные ядра, применяемые в операции замыкания с последующим открытием, имеют ширину (в пикселях), равную нечетному числу, наиболее приближенному к ширине изображения страницы (в пикселях), разделенному на 50 пикселей, при этом ширина составляет не менее 11 пикселей. Как правило, в одном из вариантов реализации вертикальное ядро, применяемое в операции замыкания с последующим открытием, имеет высоту (в пикселях), равную нечетному числу, наиболее приближенному к высоте изображения страницы (в пикселях), разделенной на 200 пикселей, при этом высота составляет не менее 11 пикселей.

На Фиг. 23A-D показано формирование сегментов изогнутой текстовой строки из центроидов блоков фрагментов. На Фиг. 23A показан закрашенный контур 2302, содержащий два блока фрагмента 2304 и 2306. Каждый блок фрагментов соотносится с центроидом 2308 и 2310 соответственно. Как показано на Фиг. 23B, из каждого центроида блока фрагмента при возможности строится исходная часть сегмента изогнутой строки текста. Данный сегмент включает первую часть 2312 и вторую часть 2314. Построение данных частей сегмента описывается ниже со ссылкой на Фиг. 24A-B. Как правило, данный сегмент строится в двух направлениях, при этом направления соотнесены с углом наклона для центроида на основе модели  $f(x,y)$ . Две исходные части 2312 и 2314 сегмента имеют длину, основанную на окружности 2316 радиуса  $r$  2318 с центром в центроиде. Как показано на Фиг. 23C, исходные части 2312 и 2314 сегмента продлеваются, так чтобы они включали дополнительные части сегмента 2318 и 2320 длиной (опять же), основанной на окружностях радиуса  $r$  вокруг конечных точек предыдущих добавленных к сегменту частей 2312 и 2314 сегмента. Как показано на Фиг. 23D, процесс построения сегмента продолжается до тех пор, пока сегмент больше не может быть продлен ни в одном направлении, с получением конечного построенного сегмента 2322, который добавляется к сегментам, построенным для центроидов блоков

фрагментов. Данные сегменты дают точки данных об углах наклона, которые могут использоваться в последующей операции согласования данных с получением более точной модели  $f(x,y)$  для углов наклона в точках на содержащем текст изображении. В одном из вариантов реализации длиной  $g$  является максимальное значение из ширины изображения страницы (в пикселях), разделенная на 100 пикселей, или 5 пикселей.

Следует отметить, что применительно к чертежам и описаниям, если не указано иное, координатами  $x$  и  $y$  являются координаты пикселей на изображении страницы. Кроме того, обычно доступны бинарные, полутоновые и исходные версии изображения страницы, которые при необходимости могут использоваться одновременно. Следует отметить, что модель  $f(x,y)$  формально возвращает тангенс угла наклона в местоположении  $(x,y)$ , однако неформально говорят, что она возвращает угол наклона, так как угол наклона легко получить из тангенса угла наклона.

На Фиг. 24А-В показано построение части сегмента на текущем конце изогнутого сегмента текстовой строки. Как показано на Фиг. 24А пиксель 2402 представляет собой текущую конечную точку сегмента изогнутой текстовой строки или центроид блока фрагмента в ходе построения одной из двух частей сегмента исходной изогнутой текстовой строки. Сначала из данной конечной точки с углом наклона  $\alpha$ , вычисленного с использованием модели  $f(x,y)$  2406, строится исходный вектор длиной  $g$  2404.

Вертикальная колонка пикселей полутонового изображения страницы 2408 в закрашенном контуре, на котором находится конец данного исходного вектора, затем используется для уточнения угла данного вектора. В примере, показанном на Фиг. 24А, исходный вектор находится на пикселе 2410 в колонке 2408. Затем рассматривается окрестность данного пикселя, включающая верхнюю окрестность 2412, нижнюю окрестность 2414 и сам пиксель 2410. Как показано на Фиг. 24В, исходный вектор 2404 может быть заменен на окончательный вектор 2420, который соединяет пиксель 2402 конечной точки с пикселем 2422 в окрестности, построенной вокруг конечной точки 2410 начального этапа, с наиболее низким значением яркости, если исходный вектор уже не соответствует окончательному вектору. Согласно правилу шкалы полутонов, чем ниже значение яркости, тем темнее пиксель. Таким образом, тогда как для определения исходного вектора используется модель  $f(x,y)$ , данный исходный вектор может быть заменен другим окончательным вектором 2420, если значение яркости пикселя, соответствующего концу исходного вектора, не является наиболее низким в окрестности, построенной вокруг конечной точки начального этапа. Другими словами, способ построения вектора предполагает, что данная модель не совсем точна, но достаточно точна для выбора некоего диапазона направлений вектора. Значения яркости в вертикальной колонке пикселей в построенной окрестности затем используются для определения определенного направления вектора из некоего диапазона направлений вектора, предоставленного моделью  $f(x,y)$ . Окончательный вектор определяет конечную точку следующей части сегмента изогнутой текстовой строки, которая продлевает сегмент изогнутой текстовой строки. Сегменты завершаются, когда способ, показанный на Фиг. 24А-В, больше не позволяет найти следующую конечную точку для следующей части сегмента. В альтернативных вариантах реализации конечная точка для следующей части сегмента может быть выбрана из подмножества пикселей с наиболее низкой яркостью в вертикальной окрестности.

Кривые текстовых строк, построенные с помощью способа, описанного со ссылкой на Фиг. 22-24В, дают более надежные точки данных для уточнения модели  $f(x,y)$ . В данной точке множество точек данных, основанных на построенных кривых текстовых строк, используется в другом раунде согласования данных с моделью  $f(x,y)$  с получением

окончательной модели, которая вычисляет тангенс углов наклона  $\alpha$  для каждого пикселя на изображении страницы. При согласовании данных снова используется любая из различных моделей согласования данных, таких как подход к согласованию данных, показанных на Фиг. 19А-В. Данными для согласования данных могут быть как раз те пиксели, которые совпадают с кривыми текстовых строк, либо кривые текстовых строк могут быть продлены в рамках закрашенных контуров текстовых строк, либо углы наклона могут быть вычислены для пикселей вне кривой путем интерполяции. Во всех случаях измеренные данные об углах наклона после построения кривых текстовых строк, как правило, дают более надежную базу для формирования модели.

Имея в распоряжении точную модель  $f(x,y)$ , ее можно использовать, по крайней мере, теоретически, для создания карты углов наклона для изображения страницы. В карте углов наклона с каждым пикселем или в некоторых случаях с ячейками, построенными на более высоких уровнях дробления, чем уровень пикселей, соотносится некий угол наклона  $\alpha$ . В настоящем описании этапов удобно использовать допущение наличия карты углов наклона, тем не менее, карта углов наклона представляет собой данные, формируемые уточненной окончательной моделью данных  $f(x,y)$ , которые также могут формироваться «на ходу», а не предварительно вычисляться и храниться в карте углов наклона.

Как показывает практика, область изображения страницы, параллельная границам корешка двух страниц разворота книги и примыкающая к ним, подвержена значительно большему перспективному искажению и искривлению страницы, чем остальные части изображения страницы. На Фиг. 25 показаны особенно сильно искривленные области изображений страниц и вычисление коэффициентов корректировки. На Фиг. 25 показаны два изображения страницы 2502 и 2504 под соответствующим изображением разворота книги 2506. Особенно сильно искаженные области 2508 и 2510 двух изображений страниц 2502 и 2504 находятся на узкой полосе пикселей шириной  $\frac{w}{12}$ . Для исправления дополнительного перспективного искажения на данных узких полосках углы наклона, вычисленные моделью  $f(x,y)$  для пикселей в данных узких областях, изменяются путем умножения на коэффициент левой страницы 2512 и коэффициент правой страницы 2514. Данные коэффициенты имеют значения, которые зависят от горизонтального положения пикселя в узкой области, при этом значение коэффициента увеличивается с 1,0 вдоль пунктирных линий 2516 и 2518 до 2,0 вдоль прилегающей к корешку границе изображения. Таким образом, модель  $f(x,y)$  и (или) исходная карта углов наклона изменяется, чтобы принять во внимание особенно сильно искаженные области 2508 и 2510.

На Фиг. 26А-Г показано выпрямление изгибов текстовых строк на изображении страницы для получения прямолинейных текстовых строк с использованием карты углов наклона, в которой с каждым пикселем на изображении страницы соотнесен некий угол наклона. На Фиг. 26А показана информация, получаемая в результате выполнения описанных ранее этапов обработки. Данная информация включает выровненное изображение 2602 страницы, как в оттенках серого, так и в цвете, если исходное изображение было получено в цвете, а также карту углов наклона 2604. Эта информация используется для формирования исправленного изображения, в котором текстовые строки и знаки и символы в текстовых строках имеют нормальную организацию плоской прямоугольной страницы. Как показано на Фиг. 26В, на заключительных этапах выпрямления искривленных текстовых строк информация о яркости и координаты пикселя на изображении 2606 страницы и угол наклона для

данного пикселя а 2608 совмещаются 2610 вместе с получением координат нового положения пикселя 2612 на исправленном изображении. Этот пиксель затем переносится или копируется на данное новое положение 2614 на исправленном изображении. Перенос пикселя означает, что значение яркости данного пикселя копируется на новое

5 местоположение на исправленном изображении страницы.

На Фиг. 26С-F приведено более подробное описание того, как изображение страницы и карта углов наклона используются для определения положений на исправленном изображении, на которое переносятся пиксели из изображения страницы. Как показано на Фиг. 26С, на первом этапе осуществляется вычисление локальных смещений  $\Delta x_i$  и

10  $\Delta y_i$  каждой ячейки  $g$  в прямоугольной сетке, наложенной на сетку пикселей изображения страницы. Прямоугольная сетка 2620 обычно имеет ячейки шириной и высотой  $g$ , где  $g$  - это четный делитель ширины и высоты пикселя. Другими словами, локальные смещения вычисляются на более высоком уровне дробления, чем сетка пикселей.

Конечно, в альтернативных вариантах реализации вместо этого для пикселей могут

15 вычисляться локальные смещения. На Фиг. 26С показан квадратный пиксель 2622 с более темными линиями границ, чем линии границ ячеек в прямоугольной сетке 2620. Смещения для ячейки  $i_{x,y}$  2624 вычисляются в примере, показанном на Фиг. 26С. Ячейка 2624 имеет координаты  $(x_i, y_i)$  по отношению к сетке ячеек 2620 и координаты  $(x', y')$  по

20 отношению к сетке ячеек в пределах области пикселя 2622. Угол наклона  $\alpha_{x_i, y_i}$  ячейки 2624 может быть вычислен непосредственно с помощью модели 2626, либо, в качестве альтернативного варианта, ей может быть присвоен угол наклона пикселя 2622, внутри которого она находится. Промежуточное значение  $\Delta y'$  для  $\Delta y_i$  вычисляется как  $g \tan$

25  $\alpha_{x_i, y_i}$  2628. Наконец, функция  $k(\ )$ , предоставляемая вместе с аргументом  $x_i = gx + x'$  2630, вычисляет смещение  $\Delta x_i$ , и с помощью значения, возвращаемого функцией  $k(\ )$  и

умноженного на промежуточное значение  $\Delta y'$ , которое было вычислено из угла наклона 2632, получается смещение  $\Delta y_i$ . Вычисления, показанные на Фиг. 26С, повторяются по

30 каждой ячейке в прямоугольной сетке ячеек 2620 с формированием карты смещений, при этом каждой ячейке  $i$  в прямоугольной сетке ячеек присваивается два смещения:  $\Delta x_i$  и  $\Delta y_i$ .

На Фиг. 26D показан один из вариантов реализации функции  $k(\ )$ , используемой для

35 формирования смещений ячеек, как описано со ссылкой на Фиг. 26С. Как показано на Фиг. 26D, функция  $k$  вычисляет значение  $k(\ )$  для ячейки  $i$  2636. Данная ячейка находится в колонке ячеек 2638, проходящей по всему изображению страницы. Верхняя ячейка

2640 и нижняя ячейка 2642 данной колонки используются функцией  $k(\ )$  для вычисления значения, возвращаемого функцией  $k(\ )$ . Углы наклона, вычисленные для данных верхней и нижней ячеек, используются для построения сходящихся векторов  $a$  и  $b$  2644-2645 или

40 сходящихся векторов 2646 и 2647, которые формируют треугольник на правой стороне колонки ячеек, соответственно, с углом  $\beta$  2648. Для правостороннего либо левостороннего треугольника с помощью выражения 2650 вычисляется косинус данного

угла. Следует отметить, что в показанном примере  $\alpha_{top}$  является отрицательным углом, а  $\alpha_{bottom}$  является положительным углом. Правосторонний треугольник подходит для

45 левой страницы разворота книги, а левосторонний треугольник - для правой страницы разворота книги. Затем функция  $k(\ )$  возвращает дробь ширины  $g$  ячейки, разделенной на ширину  $P$  пикселя, умноженную на максимальное значение из косинуса угла  $\beta$  и 0,4

(2652 на Фиг. 26D). Значение, возвращаемое функцией  $k(\ )$ , отражает степень сжатия

символов в изогнутых текстовых строках, которое наиболее выражено в наиболее изогнутых частях текстовых строк, расположенных ближе всего к границам страницы. Таким образом, смещения  $\Delta x_i$  и  $\Delta y_i$  вычисленные на Фиг. 26С, учитывают как

5 Если векторы не сходятся, функция возвращает  $1/g$ .

На Фиг. 26Е показано, как вычисленные смещения используются для переноса пикселя в исходном местоположении на изображении страницы на, как правило, другое местоположение на исправленном изображении. Прямоугольник 2656 представляет собой изображение страницы, а прямоугольник 2658 - исправленное изображение. На

10 изображении 2656 страницы выбирается одна или более опорных точек 2660-2662 и соответствующие опорные точки 2664-2666 на исправленном изображении 2658. Опорные точки - это точки, для которых могут быть вычислены координаты на исправленном изображении. В качестве одного из возможных примеров в качестве опорной точки может быть выбрана центральная точка на изображении страницы

15 возле определенного знака или символа на исходном изображении, так как можно легко найти соответствующую ей точку на исправленном изображении. В примере, показанном на Фиг. 26Е, опорная точка 2660 имеет координаты  $X, Y$ , а соответствующая опорная точка 2664 имеет координаты на исправленном изображении  $(X', Y')$ .

Рассмотрим пиксель 2668 с координатами  $(x, y)$  справа от опорной точки 2660. Для пикселя 2668 с помощью выражения 2670 могут быть вычислены относительные

20 координаты  $x^r$  и  $y^r$ . Это относительные координаты данного пикселя по отношению к опорной точке на исправленной изображении. Из-за вызванного искривлением сжатия пиксель необходимо переместить правее. Затем пиксель 2668 помещается в свое

25 положение 2672 на исправленном изображении относительно опорной точки 2664 на исправленном изображении в координатах  $(X'+x^r, Y'+y^r)$ . В выражениях 2670 относительное положение пикселя по отношению к опорной точке на исправленном

изображении  $(x^r, y^r)$  вычисляется как относительное положение данного пикселя по отношению к опорной точке на изображении страницы 2656  $(x-X, y-Y)$  плюс смещение

30  $(\Delta x, \Delta y)$ . Вычисление данных смещений показано в нижних частях Фиг. 26Е 2674. Две опорные точки 2660 и 2664 показаны на сетке 2676 ячеек большей степени дробления, описанной выше со ссылкой на Фиг. 26С. Смещение  $\Delta x$  2678 вычисляется как сумма локальных смещений, соотнесенных с каждой из ячеек между опорной точкой 2660 и опорной точкой 2664. Смещение  $\Delta y$  2680 вычисляется аналогичным образом. Для

35 пикселей слева от опорной точки в формуле для  $x^r$   $\Delta x$  вычитается, а не прибавляется.

На Фиг. 26F показан процесс переноса пикселя в целом. Из каждой опорной точки 2660-2662 на изображении страницы пиксели переносятся из указывающих направо 2682 и указывающих налево 2684 горизонтальных векторов пикселей, исходящих от

40 опорной точки, на исправленное изображение 2658 способом, описанным выше со ссылкой на Фиг. 26Е. Затем на исправленное изображение переносятся следующие пиксели выше 2680 и ниже 2688 в опорной точке и затем переносятся горизонтальная строка пикселей, включающая эти следующие два пикселя, способом, описанным выше со ссылкой на Фиг. 26Е. Данный подход применяется ко всем опорным точкам 2660-

45 2662 до тех пор, пока не будут перенесены все пиксели. В качестве альтернативного варианта может использоваться одна опорная точка, и перенос пикселей из изображения страницы на исправленное изображение может осуществляться исходя из векторов от единственной опорной точки к каждому из пикселей на изображении страницы, при этом смещения вычисляются вдоль векторов положения пикселей с учетом длины пути

грубых ячеек сетки ячеек. Как правило, карта смещений, будь то фактически сформированная или рассматриваемая теоретически, содержит достаточную информацию для переноса пикселей из изображения страницы в исправленное изображение, так чтобы в результате данного переноса перспективные искажения текстовых строк на изображении страницы были исправлены на исправленном изображении, при этом символы и знаки на исправленном изображении имели нормальную организацию прямоугольной матрицы с прямыми неизогнутыми текстовыми строками.

Иллюстрация блок-схемы одного из вариантов реализации способов и систем настоящего изобретения

На Фиг. 27A-S представлены блок-схемы, иллюстрирующие один из вариантов реализации способа корректировки перспективного искажения текстовых строк на изображениях текста, снятых мобильными устройствами и другими видами оборудования и приборов получения изображения текста. Данный способ может быть реализован на базе мобильных вычислительных приборов, в том числе смартфонов, а также на других вычислительных платформах с целью обработки содержащих текст изображений при их подготовке к оптическому распознаванию символов. Следует вновь подчеркнуть, что команды в машинном коде, которые вместе представляют собой вариант реализации данного способа в определенных устройствах, приборах и вычислительных платформах, являются полностью иллюстрированными физическими компонентами данных устройств, приборов и вычислительных платформ, в которые они включены. Хранящиеся в физической памяти команды в машинном коде представляют собой большинство систем управления машинами в современном мире, вытеснив устаревшие схемные и электромеханические системы управления. Сейчас трудно найти электронное устройство, которое работало бы на каком-либо функциональном уровне без хранящихся в его памяти команд в машинном коде, выполняемых на одном или более микропроцессорах или процессорах. Способы настоящего изобретения, реализованные в различных видах устройств, приборов и вычислительных платформ, тесно интегрированы с аппаратным обеспечением получения цифровых изображений и компонентами управления, модулями физической памяти и многими другими внутренними компонентами данных устройств, приборов и вычислительных систем и могли бы выполняться вручную или в любом другом контексте, помимо таких устройств, приборов и вычислительных систем, в которые они включены.

На Фиг. 27A приведена блок-схема одного из вариантов реализации общего способа обработки изображений страниц книг и других содержащих текст изображений. На этапе 2702a осуществляется получение изображения разворота книги вместе со ссылками на место хранения изображений двух страниц P1 и P2. На этапе 2702b вызывается подпрограмма «начальная обработка», которая осуществляет операции удаления шума, определения границ, определения контуров страниц и вертикального выравнивания границ двух изображений страниц, созданных из полученного изображения разворота книги, как описано выше со ссылкой на Фиг. 14A-15B. В случае неудачи начальной обработки, что определяется на этапе 2702c, на этапе 2702d возвращается значение false (ложь). В противном случае в цикле *for* на этапах 2702e-2702i каждое из изображений страницы P1 и P2, сформированных подпрограммой начальной обработки, обрабатываются путем вызова подпрограммы «обработать страницу» на этапе 2702f. Способ возвращает значение, возвращенное подпрограммой «обработать страницу» на этапе 2702h либо на этапе 2702j. Согласно варианту реализации настоящего

изобретения для того, чтобы было возвращено значение true (истина), необходимо, чтобы обе страницы на изображении двух страниц были обработаны успешно. В альтернативных вариантах реализации для определения успешности обработки любой одной из двух страниц в случае, если оба изображения страниц были обработаны  
5 безуспешно, может использоваться более сложное возвращаемое значение.

На Фиг. 27В показана блок-схема подпрограммы «начальная обработка», вызываемой на этапе 2702b на Фиг. 27А. На этапе 2704a подпрограмма получает изображение книги и ссылки на участки памяти, отведенные для изображений страницы P1 и P2. Если полученное изображение является цветным, что определяется на этапе  
10 2704b, данное цветное изображение преобразуется на этапе 2704c в монохромное изображение способом, описанным выше со ссылкой на Фиг. 6. На этапе 2704d данное изображение сглаживается с помощью Гауссова ядра или медианного фильтра, описанных выше со ссылкой на Фиг. 10-12. На этапе 2704e из сглаженного изображения формируется карта градиентов, как описано выше со ссылкой на Фиг. 14А-С. Карта градиентов фильтруется с помощью методики подавления немаксимальных величин, описанной выше со ссылкой на Фиг. 14В, а также путем фильтрации установленных границ, исходя из длины и других геометрических параметров с получением карты границ на этапе 2704f. На этапе 2704g из границ в карте границ, сформированной на этапе 2704 с использованием дополнительных характеристик страниц книги,  
15 распределения цветов на изображении и других геометрических данных строится контур страницы. Если на изображении не удастся установить две страницы, что определяется на этапе 2704h, то на этапе 2704i возвращается значение false (ложь). В противном случае на этапе 2704j каждое из подизображений страницы, установленных на этапе 2704g, преобразуется в изображение страницы P1 и P2, затем на этапе 2704k изображения  
20 вращаются, так чтобы прилегающая к корешку граница совпадала с вертикальной осью изображения страницы. Затем в цикле *for* этапов 2704l-2704n вызывается подпрограмма «выровнять изображение страницы» на этапе 2704m, которая осуществляет обработку каждого из двух изображений страницы P1 и P2.

На Фиг. 27С приведена блок-схема подпрограммы «выровнять изображение  
30 страницы», вызываемой на этапе 2704m на Фиг. 27В. Данная подпрограмма выполняет процедуру, описанную выше со ссылкой на Фиг. 15В. На этапе 2706a подпрограмма определяет угол наклона  $\alpha$  не прилегающая к корешку граница страницы. Если  $\alpha$  меньше порогового угла, что определяется на этапе 2706b, то дальнейшая обработка не требуется и на этапе 2706c подпрограмма завершается. В противном случае на этапе  
35 2706d подпрограмма определяет точку пересечения между границей страницы, параллельной корешку, но не прилегающей к нему, и верхней границей страницы (точка 1522 на Фиг. 15В). На этапе 2706e подпрограмма определяет ширину и высоту изображения страницы в пикселях. В цикле *for* этапов 2706f-2706n осуществляется обработка каждого горизонтального ряда пикселей на части изображения страницы, которая представляет собой данную страницу. На этапе 2706g определяется точка, в которой граница страницы, параллельная корешку, но не прилегающая к нему,  
40 пересекается с данным рядом пикселей (1524 на Фиг. 15В). Затем на этапе 2706i или 2706j определяются длины трех сегментов горизонтального ряда, описанных со ссылкой на Фиг. 15В. На этапе 2706k определяется число пикселей, подлежащих удалению из области страницы. Затем на этапе 2706l и 2706m дублируются определенные пиксели в не относящейся к странице части горизонтального ряда пикселей и удаляются определенные пиксели в относящейся к странице части горизонтального ряда пикселей, как показано на Фиг. 15В.

На Фиг. 27D приведена блок-схема подпрограммы «обработать страницу» вызываемой на этапе 2702f на Фиг. 27A. На этапе 2708a подпрограмма получает ссылку на изображение страницы. На этапе 2708b вызывается подпрограмма «разделить фрагменты слов на блоки», которая выполняет способ разделения фрагментов на блоки, описанный выше со ссылкой на Фиг. 16A-17D. Если способ разделения фрагментов на блоки выполнен безуспешно, что определяется на этапе 2708c, подпрограмма на этапе 2708d возвращает значение false (ложь). В противном случае на этапе 2708e вызывается подпрограмма «исходная модель», которая создает исходную модель  $f(x,y)$  путем согласования данных, как описано выше со ссылкой на Фиг. 19A-B. Если модель не сходится, что определяется на этапе 2708f, то при доступности другой модели согласования данных, что определяется на этапе 2708g, этап формирования исходной модели 2708e повторяется с использованием другой модели. В противном случае подпрограмма на этапе 2708h возвращает значение false (ложь). На этапе 2708i, если исходная модель была создана, вызывается подпрограмма «фильтровать модель», которая фильтрует блоки фрагментов слов, как описано выше со ссылкой на Фиг. 20. В случае безуспешной фильтрации, что определяется на этапе 2708j, подпрограмма возвращает значение false (ложь) на этапе 2708k. В противном случае на этапе 2708l вызывается подпрограмма «выровнять страницу», которая выполняет оставшиеся этапы способа настоящего изобретения.

На Фиг. 27E приведена блок-схема подпрограммы «разделить фрагменты слов на блоки», вызываемой на этапе 2708b на Фиг. 27D. На этапе 2710a вызывается подпрограмма, которая инициализирует два множества больших двоичных объектов (блобов) или скоплений символьных пикселей  $bSet$  и  $xSet$ . На этапе 2710b переменной  $progress$  (ход выполнения) задается значение false (ложь). В цикле  $for$  этапов 2710c-2710g на этапе 2710d осуществляется обработка текущего множества блобов или скоплений распознанных символьных пикселей, с тем чтобы найти любые новые блобы или скопления символьных пикселей на изображении страницы, путем вызова подпрограммы «новые блобы». В случае нахождения новых блобов, что определяется на этапе 2710e, на этапе 2710f переменной  $progress$  (ход выполнения) присваивается значение true (истина). Выполнение данного процесса продолжается во внешнем цикле этапов 2710b-2710h до тех пор, пока не удастся определить ни одного дополнительного блоба. Процесс определения и пополнения множества блобов является одним из возможных вариантов реализации способа настоящего изобретения, описываемого в общих чертах со ссылкой на Фиг. 16C-G. На этапе 2710i множеству блоков фрагментов  $fSet$  отводятся пространства, а на этапе 2710j вызывается подпрограмма «обработать блобы», которая создает блоки фрагментов из блобов, как описано выше со ссылкой на Фиг. 17A-D. На этапах 2710k и 2710l возвращается значение, возвращенное подпрограммой «обработать блобы». Также на этапе 2710l также возвращается ссылка на множество блоков фрагментов  $fSet$ .

На Фиг. 27F приведена блок-схема подпрограммы инициализации, вызываемой на этапе 2710a на Фиг. 27E. На этапе 2712a создается множество  $pSet$  пикселей, относительно равномерно распределенных по изображению страницы, и множествам  $bSet$  и  $xSet$  блобов отводятся участки памяти. В цикле  $for$  этапов 2712b-2712h подпрограмма пытается расширить каждый пиксель во множестве  $pSet$  до блоба или скопления символов. Если пиксель  $p$  еще не является членом блоба, что определяется на этапе 2712c, на этапе 2712d вызывается подпрограмма «расширить пиксель», которая пытается расширить пиксель до скопления символьных пикселей или блоба. Если блоб, возвращенный подпрограммой «расширить пиксель», больше порогового размера,

что определяется на этапе 2712e, то на этапе 2712f данный блок добавляется к множеству bSet. В противном случае на этапе 2712g блок добавляется во множество xSet. Таким образом, во множество bSet добавляются только блоки размера, достаточного для того, чтобы составлять фрагменты слов.

5 На Фиг. 27G показано, как в одном из вариантов реализации определяется, что пиксель является символьным. Если значение яркости пикселя ниже порогового значения, что определяется на этапе 2714a, пиксель считается символьным, и на этапе 2714b возвращается значение true (истина). В противном случае на этапе 2714c возвращается значение false (ложь), означающее, что пиксель не является символьным.

10 В других вариантах реализации для определения того, является ли некий пиксель символьным, могут применяться дополнительные аспекты, в том числе исследование окрестности пикселя, на предмет того, является ли пиксель частью скопления пикселей. Так как все еще доступно цветное изображение (если исходное изображение является цветным), то для отличия символьных пикселей от несимвольных фоновых пикселей

15 могут применяться различные виды связанных с цветом аспектов. Конечно, страницы текста также могут содержать чертежи, схемы и прочие нетекстовые области, поэтому для отбраковки пикселей из данных видов областей могут использоваться дополнительные аспекты. Тем не менее, как описано ниже, блоки, изначально полученные в рамках способа настоящего изобретения, фильтруются, с тем чтобы

20 оставить только блоки, которые с высокой степенью вероятности составляют фрагменты слов.

На Фиг. 27H приведена блок-схема подпрограммы «расширить пиксель», вызываемой на этапе 2712d на Фиг. 27F. На этапе 2716a подпрограмма получает координаты некоего пикселя. На этапе 2716b назначается блок b и его исходный член принимает значение

25 полученного пикселя p. В цикле while этапов 2716e в блок добавляются символы, прилегающие к исходному пикселю p, а также любые символы, добавленные в рамках предыдущих итераций цикла while.

На Фиг. 27I приведена блок-схема подпрограммы «новые блоки», вызываемой на этапе 2710d на Фиг. 27E. На этапе 2718a подпрограмма получает блок b и ссылки на множества блоков bSet и xSet. На этапе 2718b локальная переменная pSet

30 устанавливается так, чтобы она включала группу равноудаленных пикселей на периферии блока b с расстоянием между ними в не менее чем 1,5 раза больше длины  $r_1$ , описанной выше, и локальная переменная found (найдено) принимает значение false (ложь). Затем в цикле for этапов 2718c-2718j из каждого из пикселей во множестве pSet

35 осуществляется поиск дополнительных символьных пикселей, еще не включенных в какой-либо блок в пределах расстояния  $r_1$  пикселя. Если такой поиск позволяет найти такой новый пиксель q, что определяется на этапе 2718e, то данный пиксель расширяется путем вызова подпрограммы «расширить пиксель» на этапе 2718f, а блок, возвращенный подпрограммой, добавляется на одном из этапов 2718h и 2718i либо во множество bSet,

40 либо во множество xSet в зависимости от того, превышает блок пороговый размер, что определяется на этапе 2718g.

На Фиг. 27J приведена блок-схема подпрограммы «обработать блоки», вызываемой на этапе 2710j на Фиг. 27E. На этапе 2720a подпрограмма получает пустое множество fSet блоков фрагментов и множество bSet блоков. Затем в цикле for этапов 2720b-2720g рассматривается каждый блок во множестве bSet. Если размер рассматриваемого в

45 текущий момент блока больше порогового, что определяется на этапе 2720c, то на этапе 2720d определяется центроид и угол наклона  $\alpha$  данного блока, как описано со ссылкой на Фиг. 17A-C. Затем на этапе 2720e вокруг данного блока строится

параллелограмм или прямоугольник наименьших возможных размеров, как описано выше со ссылкой на Фиг. 17D. На этапе 2720f блок фрагмента, полученный на этапе 2720e, добавляется во множество  $fSet$ . На этапе 2720h блоки фрагментов фильтруются, и значение, возвращенное подпрограммой фильтрации, вызванной на этапе 2720h,

5 возвращается подпрограммой «обработать блобы» на одном из этапов 2720i и 2720j. На Фиг. 27K приведена блок-схема подпрограммы «фильтровать блоки фрагментов», вызываемой на этапе 2720h на Фиг. 27J. На этапе 2722a подпрограмма получает множество блоков фрагментов  $fSet$ . На этапе 2722b назначается пустое множество блоков фрагментов  $tSet$  и устанавливается на значение пустого множества. В цикле *for* 10 этапов 2722c-2722j в отношении каждого фрагмента, содержащегося во множестве  $fSet$ , применяются различные тесты, и только если рассматриваемый в текущий момент фрагмент успешно проходит все тесты, он помещается во множество  $tSet$  на этапе 2722L. Если число блоков фрагментов во множестве  $tSet$  меньше порогового, что определяется на этапе 2722k, подпрограмма на этапе 2722l возвращает значение *false* (ложь). В 15 противном случае  $fSet$  на этапе 2722 т принимается равным  $tSet$ , и на этапе 2722n возвращается значение *true* (истина). В число различных аспектов фильтрации входят определение на этапе 2722d того, превышает ли угол наклона  $\alpha$  блока фрагмента пороговое значение, и определение на этапе 2722e-2722h, не выходят ли ширина и длина блока фрагмента за пределы приемлемых диапазонов. Многоточие 2722o указывает на то, что в рамках операции фильтрации могут применяться и другие тесты. 20

На Фиг. 27L показана блок-схема подпрограммы «фильтровать модель», вызываемой на этапе 2708i на Фиг. 27D. Данная подпрограмма эквивалентна способу фильтрации, описанному выше со ссылкой на Фиг. 20. Данная подпрограмма также подобна подпрограмме, описанной выше со ссылкой на Фиг. 27K. В цикле *for* этапов 2724a-d 25 блоки фрагментов отвергаются, если значение угла наклона  $\alpha$  недостаточно близко к значению угла наклона  $\alpha$ , вычисленному моделью  $f(x,y)$ . Если значение наблюдаемого угла наклона  $\alpha$  отличается от угла наклона  $\alpha$ , вычисленного из исходной модели  $f(x,y)$  менее чем на пороговую величину, блок фрагмента копируется во множество  $tSet$ . Если любой из блоков фрагментов был удален, что определяется на этапе 2724e, то на этапе 30 2724f исходная модель вычисляется заново. В альтернативных вариантах реализации исходная модель вычисляется заново, если удаленных блоков фрагментов в ходе фильтрации было удалено больше порогового значения. Если размер множества  $tSet$  меньше порогового значения, что определяется на этапе 2724g, то подпрограмма «фильтровать модель» на этапе 2724h возвращает значение *false* (ложь). В противном 35 случае на этапе 2724i,  $fSet$  принимается равным  $tSet$ , и на этапе 2724j возвращается значение *true* (истина).

На Фиг. 27M приведена блок-схема подпрограммы «выровнять страницу», вызываемой на этапе 2708l на Фиг. 27D. На этапе 2726a подпрограмма «выровнять страницу» получает множество блоков фрагментов слов  $fSet$ , текущую модель 40 искривлений  $f(x,y)$  и изображение страницы. Кроме того, подпрограмма «выровнять страницу» назначает четыре карты страницы: (1) *initialTextMap*, структура данных, которая содержит закрашенные контуры, созданные способом, описанным со ссылкой на Фиг. 22; (2) *segmentMap*, структура данных, которая содержит изогнутые сегменты текстовых строк, созданные способом, описанным выше со ссылкой на Фиг. 23A-24B; 45 (3) *angleMap*, структура данных, которая содержит карту углов наклона, описанную выше со ссылкой на Фиг. 25; и (4) *offsetMap*, структура данных, которая содержит карту смещений, описанную выше со ссылкой на Фиг. 26C. Как было описано ранее, данные карты в определенных вариантах реализации могут быть структурами фактических

данных, тогда как в других они могут быть теоретическими структурами, которые не хранятся на физических носителях, а создаются «на ходу» из модели данных и другой информации. На этапе 2726b вызывается подпрограмма «исходный текст», которая подготавливает карту текста способом, описанным выше со ссылкой на Фиг. 22. На этапе 2726c вызывается подпрограмма «сегменты текста», которая выполняет способ, описанный выше со ссылкой на Фиг. 23А-24В. На этапе 2726d вызывается подпрограмма «окончательная модель», которая выполняет финальный этап согласования данных, на котором на основе содержимого карты векторов, сформированной на этапе 2726c, формируется окончательная модель  $f(x,y)$ . На этапе 2726e вызывается подпрограмма «карта углов», которая формирует карту углов, описанную выше со ссылкой на Фиг. 25. На этапе 2726f вызывается подпрограмма «карта смещений», которая формирует карту смещений, описанную выше со ссылкой на Фиг. 26С. Наконец, на этапе 2726g вызывается подпрограмма «завершить», которая формирует исправленное изображение из изображения страницы и карты смещений в соответствии со способом, описанным выше со ссылкой на Фиг. 26Е-Е.

На Фиг. 27N приведена блок-схема подпрограммы «исходный текст», вызываемой на этапе 2726b на Фиг. 27М. На этапе 2728a подпрограмма получает исходную карту текста и изображение страницы. На этапе 2728b символные пиксели переносятся из изображения страницы на исходную карту текста с сохранением своего местоположения. Исходная карта текста является бинарным изображением. На этапе 2728c выполняется операция замыкания, описанная выше со ссылкой на Фиг. 21А-Е, которая формирует промежуточную карту. На этапе 2728d выполняется операция закрытия, описанная выше со ссылкой на Фиг. 21А-Е, которая формирует окончательную карту текста (220b на Фиг. 22). Наконец, на этапе 2728e выполняется гауссово сглаживание в вертикальном направлении. В альтернативных вариантах реализации в отношении копии полутонового изображения страницы применяются операторы замыкания и открытия.

На Фиг. 27O приведена блок-схема подпрограммы «сегменты текста». На этапе 2730a подпрограмма «сегменты текста» получает исходную карту текста, пустую карту сегментов и множество блоков фрагментов  $fSet$ . Во внешнем цикле  $for$  этапов 2730b-2730i рассматривается каждый блок фрагментов. Во внутреннем цикле  $for$  этапов 2730c-2730g рассматривается каждое из двух направлений построения сегментов изогнутых текстовых строк, описанного выше со ссылкой на Фиг. 23В. В наиболее внутреннем цикле  $while$  этапов 2730d-2730f формируется одна сторона сегмента изогнутой текстовой строки путем последовательного добавления частей сегментов, как описано выше со ссылкой на Фиг. 23А-24В. На этапе 2730h по завершению формирования следующего сегмента изогнутой текстовой строки сегмент добавляется в карту сегментов.

На Фиг. 27P приведена блок-схема подпрограммы «карта углов»), вызываемой на этапе 2726e на Фиг. 27М. Данная подпрограмма на этапе 2732b вычисляет угол наклона каждого пикселя изображения страницы из окончательной модели и помещает угол наклона на карту углов в положении, соответствующем пикселю изображения страницы. На этапе 2732c углы в карте углов в узких вертикальных областях вдоль границы корешка страницы на изображении страницы умножаются на вычисленные коэффициенты, как описано выше со ссылкой на Фиг. 25.

На Фиг. 27Q приведена блок-схема подпрограммы «карта смещений», вызываемой на этапе 2726f на Фиг. 27М. На этапе 2734a подпрограмма получает ссылку на пустую карту смещений и карту углов, сформированную подпрограммой «карта углов». Как описано выше со ссылкой на Фиг. 26С, карта смещений может иметь прямоугольную сетку с более высоким уровнем разбиения, чем у сетки пикселей изображения страницы.

В цикле *for* этапов 2734b-2734h для каждой ячейки *i* на карте смещений вычисляется значение смещения. Данная операция вычисления выполняется в соответствии со способом, описанным выше со ссылкой на Фиг. 26C и 26D. На этапах 2734c-2734f для рассматриваемой в текущий момент ячейки *i* вычисляется коэффициент *k*, как описано  
 5 выше со ссылкой на Фиг. 26D. На этапе 2734g для ячейки *i* в соответствии со способом, описанным выше со ссылкой на Фиг. 26C, вычисляются локальные смещения  $\Delta x_i$  и  $\Delta y_i$ .

На Фиг. 27R приведена блок-схема подпрограммы «завершить», вызываемой на этапе 2726g на Фиг. 27M. Данная подпрограмма выполняет способ переноса пикселей, описанный на Фиг. 26E-F, для создания с помощью подпрограммы «карта смещений»  
 10 исправленного изображения из изображения страницы и карты смещений. На этапе 2736a подпрограмма получает карту изображения и карту смещений. На этапе 2736b назначается пустая страница *tPage*, и не относящиеся к странице пиксели на изображении страницы копируются в *tPage*. На этапе 2736c определяется множество опорных точек, а на этапе 2736d определяются их местоположения на изображении страницы, а также  
 15 страница *tPage*, которая будет содержать исправленное изображение страницы, как описано выше со ссылкой на Фиг. 26E. В трижды вложенных циклах *for* этапов 2736e-2736k осуществляется перенос пикселей из изображения страницы на исправленное изображение, как описано со ссылкой на Фиг. 26F. Для переноса пикселей из изображения страницы на исправленное изображение страницы каждая точка в  
 20 вертикальном диапазоне, или вертикальной окрестности, вокруг каждой опорной точки расширяется вправо и влево. Каждый пиксель переносится путем вызова подпрограммы «перенести пиксели» на этапе 2736h.

На Фиг. 27S приведена блок-схема подпрограммы «перенести пиксели», вызываемой на этапе 2736h на Фиг. 27R. На этапе 2738a подпрограмма получает ссылки на  
 25 изображение страницы и исправленное изображение *tPage*, координаты пикселя *i* на изображении страницы, координаты соответствующего пикселя *k* на исправленном изображении, направление, в котором осуществляется расширение вокруг опорной точки или точки в вертикальной окрестности опорной точки для переноса пикселя, а также карту смещений. Локальные переменные *sum\_y* и *sum\_x* на этапе 2738b принимают  
 30 значение 0. Затем в цикле *for* этапов 2738c-2738g каждый пиксель *j* в горизонтальной строке пикселей (начиная с пикселя *i* и до границы страницы) в направлении *d* переносится из изображения страницы на исправленное изображение *tPage*. На этапе 2738e вычисляется локальное смещение пикселя *j* путем добавления локальных смещений к локальным переменным *sum\_y* и *sum\_x*, а затем на этапе 2738f вычисляется положение  
 35 данного пикселя на исправленном изображении и значение яркости данного пикселя переносится из изображения страницы на исправленное изображение.

Хотя настоящее изобретение описывается на примере конкретных вариантов реализации, предполагается, что оно не будет ограничено только этими вариантами реализации. Специалистам в данной области техники будут очевидны возможные  
 40 модификации в пределах сущности настоящего изобретения. Например, любые из множества различных параметров реализации и проектирования, в том числе модульная организация, язык программирования, аппаратная платформа, структуры управления, структуры данных и прочие параметры реализации и проектирования, могут  
 45 варьироваться для получения альтернативных вариантов реализации способов и систем настоящего изобретения. Как говорилось выше, существуют альтернативные способы, которые могут использоваться для многих, если не всех, различных этапов обработки, участвующих в исправлении искривления текстовых строк. Кроме того, множество различных пороговых значений, значений ширины ядра, значения степени разбиения

карты изображения и другие параметры могут отличаться в различных вариантах реализации.

Следует понимать, что представленное выше описание вариантов реализации изобретения приведено для того, чтобы дать возможность любому специалисту в данной области техники воспроизвести или применить настоящее изобретение. Специалистам в данной области техники будут очевидны возможные модификации представленных вариантов реализации, при этом общие принципы, представленные здесь без отступления от сущности и объема настоящего изобретения, могут быть применены в рамках других вариантов реализации. Таким образом, подразумевается, что настоящее изобретение не ограничено представленными здесь вариантами реализации. Напротив, в его объем входят все возможные варианты реализации, согласующиеся с раскрываемыми в настоящем документе принципами и новыми отличительными признаками.

#### (57) Формула изобретения

1. Подсистема обработки изображений в составе устройства, прибора или системы, предназначенных для оптического распознавания текста, которая получает содержащее текст с изогнутыми текстовыми строками изображение и вводит полученное содержащее текст изображение в подсистему обработки изображений, которая, в свою очередь, формирует соответствующее исправленное изображение с выпрямленными текстовыми строками с прямоугольной организацией, при этом такая подсистема обработки изображений включает:

отдельный аппаратный уровень или аппаратный уровень, используемый совместно с устройством, прибором или системой, включающий один или более процессоров, один или более модулей памяти; и

команды в машинном коде, хранящиеся в одном или более физических устройств хранения данных, которые при их выполнении одним или несколькими из одного или более процессоров управляют работой подсистемы обработки изображений в части следующих операций:

получение введенного содержащего текст изображения;

определение на содержащем текст изображении подызображения страницы;

исправление вертикальной перспективы подызображения страницы;

формирование модели искривления текстовых строк для подызображения страницы, в которой с пикселями на подызображении страницы соотносятся углы наклона фрагментов текста;

формирование с использованием модели искривления текстовых строк локальных смещений пикселей на подызображении страницы; и

перенос значений яркости пикселей из подызображения страницы на исправленное подызображение страницы, используя смещения пикселей, сформированные из локальных смещений, для построения исправленного подызображения страницы, в котором текстовые строки выпрямлены и в котором текстовые знаки и символы имеют прямоугольное расположение.

2. Подсистема обработки изображений по п. 1, отличающаяся тем, что подсистема обработки изображений определяет на содержащем текст изображении подызображение страницы путем:

определения границ яркости на содержащем текст изображении; и

использования определенных границ яркости, а также известных геометрических данных, соотношенных с содержащими текст страницами, с целью определения подызображений, соответствующих одной или более содержащим текст страницам на

содержащем текст изображения.

3. Подсистема обработки изображений по п. 1, отличающаяся тем, что подсистема обработки изображений исправляет вертикальную перспективу подызображения страницы путем:

- 5 вращения подызображения страницы, так чтобы прилегающая к корешку граница страницы книги имела вертикальную ориентацию; и  
корректировки изображения, так чтобы не прилегающая к корешку граница также имела вертикальную ориентацию.

4. Подсистема обработки изображений по п. 1, отличающаяся тем, что подсистема обработки изображений формирует модель искривления текстовых строк для  
10 подызображения страницы, в которой с пикселями на подызображении страницы соотносятся углы наклона фрагментов текста, путем:

- разделения фрагментов текста на блоки на подызображении страницы, при этом с  
каждым фрагментом текста соотносится угол наклона;  
15 использования углов наклона, соотнесенных с фрагментами текста, для определения коэффициентов для полиномиального представления исходной модели искривления текстовых строк; и  
уточнения исходной модели искривления текстовых строк для получения окончательной модели искривления текстовых строк.

5. Подсистема обработки изображений по п. 1, отличающаяся тем, что модель  
20 искривления текстовых строк возвращает тангенс угла наклона фрагментов текста при ее вызове с помощью аргументов, представляющих местоположение пикселя на подызображении страницы.

6. Подсистема обработки изображений по п. 4, отличающаяся тем, что подсистема  
25 обработки изображений разделяет фрагменты текста на блоки на подызображении страницы путем:

- определения фрагментов текстов на подызображении страницы, которые составляют  
скопления знаков и/или символов, которые соответствуют словам, фрагментам слов  
и/или фрагментам нескольких слов; и  
30 для каждого фрагмента текста -  
определения центроида фрагмента текста и  
применения способа минимизации, позволяющего минимизировать расстояние между  
пикселями каждого фрагмента текста и линией, проходящей через центроид, для  
определения угла наклона фрагмента текста, то есть угла между основной осью  
35 фрагмента текста и горизонтальным направлением, и его соотнесения с данным  
фрагментом текста.

7. Подсистема обработки изображений по п. 6, отличающаяся тем, что подсистема  
обработки изображений определяет фрагменты текста на подызображении страницы  
путем:

- 40 отбора с использованием отправных точек символьных пикселей на изображении  
страницы;  
поочередного расширения каждого отображенного символьного пикселя до блока  
смежных символьных пикселей, а также  
поиска с использованием отправных точек новых символьных пикселей в пределах  
45 максимального расстояния блока смежных символьных пикселей.

8. Подсистема обработки изображений по п. 6,  
отличающаяся тем, что, если содержащее текст изображение включает более темные  
знаки на более ярком фоне, символьный пиксель имеет яркость ниже порогового

значения; и

отличающаяся тем, что, если содержащее текст изображение включает более светлые знаки на более темном фоне, символьный пиксель имеет яркость больше порогового значения.

5 9. Подсистема обработки изображений по п. 4, отличающаяся тем, что после разделения фрагментов текста на блоки на подызоображении страницы подсистема обработки изображений фильтрует блоки фрагментов текста по углу наклона и размеру, с тем чтобы отобрать набор блоков фрагментов текста, наиболее подходящих для определения коэффициентов для полиномиального представления исходной модели

10 искривления текстовых строк.

10. Подсистема обработки изображений по п. 4, отличающаяся тем, что подсистема обработки изображений использует углы наклона, соотнесенные с фрагментами текста, для определения коэффициентов для полиномиального представления исходной модели искривления текстовых строк путем:

15 инициализации значений множества коэффициентов для исходного полиномиального представления модели искривления текстовых строк; и

применения способа согласования данных для определения коэффициентов для исходного полиномиального представления модели искривления текстовых строк на основе углов наклона, соотнесенных с фрагментами текста.

20 11. Подсистема обработки изображений по п. 10, отличающаяся тем, что способ согласования данных выбирается из следующих способов:

метод Гаусса, применяемый к системе линейных уравнений, сформированных из производных по отношению к коэффициентам модели выражений для углов наклона фрагментов текста в квадрате;

25 метод Левенберга-Марквардта;

метод Гаусса-Ньютона;

нелинейный метод сопряженных градиентов; и

дополнительные линейные и нелинейные методы согласования искривления на основе регрессии.

30 12. Подсистема обработки изображений по п. 10, отличающаяся тем, что подсистема обработки изображений:

фильтрует блоки фрагментов текста для обеспечения соответствия углов наклона, соотнесенных с каждым блоком фрагментов текста, углом наклона, определенным для пикселей блоков фрагментов текста с использованием исходного полиномиального

35 представления модели искривления текстовых строк; и

если фильтрация блоков фрагментов текста ведет к удалению блоков фрагментов текста в количестве, превышающем пороговое значение,

заново формирует коэффициенты для исходного полиномиального представления модели искривления текстовых строк путем применения способа согласования данных.

40 13. Подсистема обработки изображений по п. 4, отличающаяся тем, что подсистема обработки изображений уточняет исходную модель искривления текстовых строк для получения окончательной модели искривления текстовых строк путем:

выполнения в отношении подызоображения страницы операций замыкания, открытия и вертикального сглаживания для формирования закрашенных контуров,

45 соответствующих текстовым строкам на подызоображении страницы;

построения в пределах закрашенных контуров вокруг текстовых строк сегментов искривленных текстовых строк с использованием углов наклона фрагментов текста, сформированных полиномиальным представлением модели искривления текстовых

строк; и

использования точек и соотнесенных с ними углов наклона фрагментов текста вдоль построенных сегментов искривленных текстовых строк в качестве данных для применения способа согласования данных для формирования уточненных коэффициентов для полиномиального представления модели искривления текстовых строк.

14. Подсистема обработки изображений по п. 13, отличающаяся тем, что подсистема обработки изображений, используя углы наклона фрагментов текста, сформированные полиномиальным представлением модели искривления текстовых строк, строит сегменты изогнутых текстовых строк в пределах закрашенных контуров вокруг текстовых строк путем:

для каждого блока фрагментов текста на подизображении страницы -

отбора в качестве отправной точки пикселя, соответствующего центру блока фрагмента текста в пределах закрашенного контура, соответствующего текстовой строке; и

построения сегмента изогнутой текстовой строки в двух направлениях от отправной точки путем итерационного добавления частей сегмента к текущим концам сегмента изогнутой текстовой строки до тех пор, пока не удастся добавить ни одной части сегмента.

15. Подсистема обработки изображений по п. 14, отличающаяся тем, что подсистема обработки изображений строит из каждой отправной точки первую часть сегмента в направлении угла наклона фрагмента текста, полученного из полиномиального представления модели искривления текстовых строк для данной отправной точки и вторую часть сегмента в направлении, равном направлению угла наклона плюс  $180^\circ$ , при этом отправная точка представляет собой две текущие конечные точки сегмента изогнутой текстовой строки.

16. Подсистема обработки изображений по п. 14, отличающаяся тем, что подсистема обработки изображений добавляет часть сегмента к текущей конечной точке сегмента изогнутой текстовой строки путем:

определения угла наклона фрагмента текста с помощью полиномиального представления модели искривления текстовых строк для пикселя, соответствующего текущей конечной точке сегмента изогнутой текстовой строки;

построения из текущей конечной точки сегмента изогнутой текстовой строки исходного вектора с направлением, равным определенному углу наклона фрагмента текста;

рассмотрения вертикальной окрестности пикселей, совпадающих с концом исходного вектора;

если некий пиксель в вертикальной окрестности имеет значение яркости, более приближенное к ожидаемой яркости текстовых пикселей, чем яркость членов подмножества оставшихся пикселей в данной вертикальной окрестности -

повторного формирования исходного вектора при необходимости для получения окончательного вектора, конец которого совпадает с пикселем, имеющим значение яркости, более приближенное к ожидаемому, и

продления сегмента изогнутой текстовой строки до пикселя, совпадающего с концом окончательного вектора, и

если ни один из пикселей в вертикальной окрестности не имеет значения яркости, более приближенного к ожидаемой яркости текстовых пикселей, чем любой член подмножества оставшихся пикселей в данной вертикальной окрестности - завершения

сегмента изогнутой текстовой строки в текущей конечной точке сегмента изогнутой текстовой строки.

17. Подсистема обработки изображений по п. 1, отличающаяся тем, что подсистема обработки изображений, используя модель искривления текстовых строк, формирует  
5 локальные смещения пикселей на подызображении страницы путем:

формирования карты углов наклона, которая включает угол наклона каждого пикселя подызображения страницы, путем ввода положения пикселя на подызображении страницы в окончательную модель искривления текстовых строк;

10 изменения углов наклона в вертикальной области карты углов путем умножения на коэффициент, который корректирует углы наклона с учетом увеличенного искривления и сжатия в соответствующей вертикальной области подызображения страницы; и

формирования карты смещений, которая включает локальное смещение каждой ячейки карты смещений на подызображении страницы.

18. Подсистема обработки изображений по п. 17, отличающаяся тем, что подсистема  
15 обработки изображений формирует локальное смещение ячейки на карте смещений путем:

определения значения вертикального смещения как коэффициента сжатия, умноженного на ширину ячейки и далее умноженного на тангенс угла наклона, полученного из окончательной модели искривления текстовых строк;

20 установки горизонтального смещения на значение коэффициента сжатия; и соотношения вертикального смещения и горизонтального смещения с ячейкой;

19. Подсистема обработки изображений по п. 18, отличающаяся тем, что коэффициент сжатия для ячейки вычисляется как дробь, равная ширине ячейки, разделенной на ширину пикселя, с умножением результата на косинус разницы между углами наклона  
25 для ячеек сверху и снизу колонки вертикальных ячеек, проходящей через изображение страницы и совпадающей с ячейкой, или на 0,4.

20. Подсистема обработки изображений по п. 1, отличающаяся тем, что подсистема обработки изображений, используя смещения пикселей, сформированные из локальных смещений, переносит значения яркости пикселей из подызображения страницы на  
30 исправленное подызображение страницы путем:

определения на подызображении страницы и исправленном подызображении страницы пар соответствующих опорных точек;

переноса значения яркости пикселей, соответствующих опорным точкам на подызображении страницы, на пиксели, соответствующие опорным точкам на  
35 исправленном подызображении страницы; и

для каждого не относящегося к опорным точкам пикселя на подызображении страницы,

40 суммирования локальных смещений, совпадающих с вектором от не относящегося к опорным точкам пикселя до ближайшей опорной точки, с получением смещения пикселя, и

переноса значения яркости не относящегося к опорным точкам пикселя на пиксель, смещенный от соответствующей опорной точки на исправленном подызображении страницы на значение смещения пикселя.

21. Способ, выполняемый подсистемой обработки изображений в составе устройства,  
45 прибора или системы, предназначенных для оптического распознавания текста, которая получает содержащее текст с изогнутыми текстовыми строками изображение и вводит полученное содержащее текст изображение в подсистему обработки изображений для создания соответствующего исправленного изображения с выпрямленными текстовыми

строками с прямоугольной организацией, при этом данный способ включает:

получение исходного содержащего текст изображения и сохранение его в памяти;  
определение на содержащем текст изображении подызображения страницы;  
исправление вертикальной перспективы подызображения страницы;

5 формирование модели искривления текстовых строк для подызображения страницы, в которой с пикселями подызображения страницы соотносятся определенные углы наклона фрагментов текста;

на основе модели искривления текстовых строк формирование из модели искривления текстовых строк локальных смещений пикселей на подызображении страницы;

10 перенос значений яркости пикселей из подызображения страницы на исправленное подызображение страницы, используя смещения пикселей, сформированные из локальных смещений, для построения в памяти исправленного подызображения страницы, в котором текстовые строки выпрямлены и в котором текстовые знаки и символы имеют прямоугольное расположение.

15 22. Физическое устройство хранения данных, которое хранит команды в машинном коде, которые при их выполнении одним или более процессорами аппаратного уровня подсистемы обработки изображений, которая включает один или более процессоров и один или более модулей памяти и которая является частью некоего устройства, прибора или системы, предназначенных для оптического распознавания текста, получающей  
20 содержащее текст с изогнутыми текстовыми строками изображение, управляют получением содержащего текст изображения от устройства, прибора или системы и формированием соответствующего исправленного изображения с выпрямленными текстовыми строками с прямоугольной организацией путем следующих операций:

получение исходного содержащего текст изображения и сохранения его в памяти;  
25 определение на содержащем текст изображении подызображения страницы;  
исправление вертикальной перспективы подызображения страницы;

формирование модели искривления текстовых строк для подызображения страницы, в которой с пикселями подызображения страницы соотносятся углы наклона фрагментов текста;

30 на основе модели искривления текстовых строк формирование из модели искривления текстовых строк локальных смещений пикселей на подызображении страницы;

перенос значений яркости пикселей из подызображения страницы на исправленное подызображение страницы, используя смещения пикселей, сформированные из локальных смещений, для построения в памяти исправленного подызображения

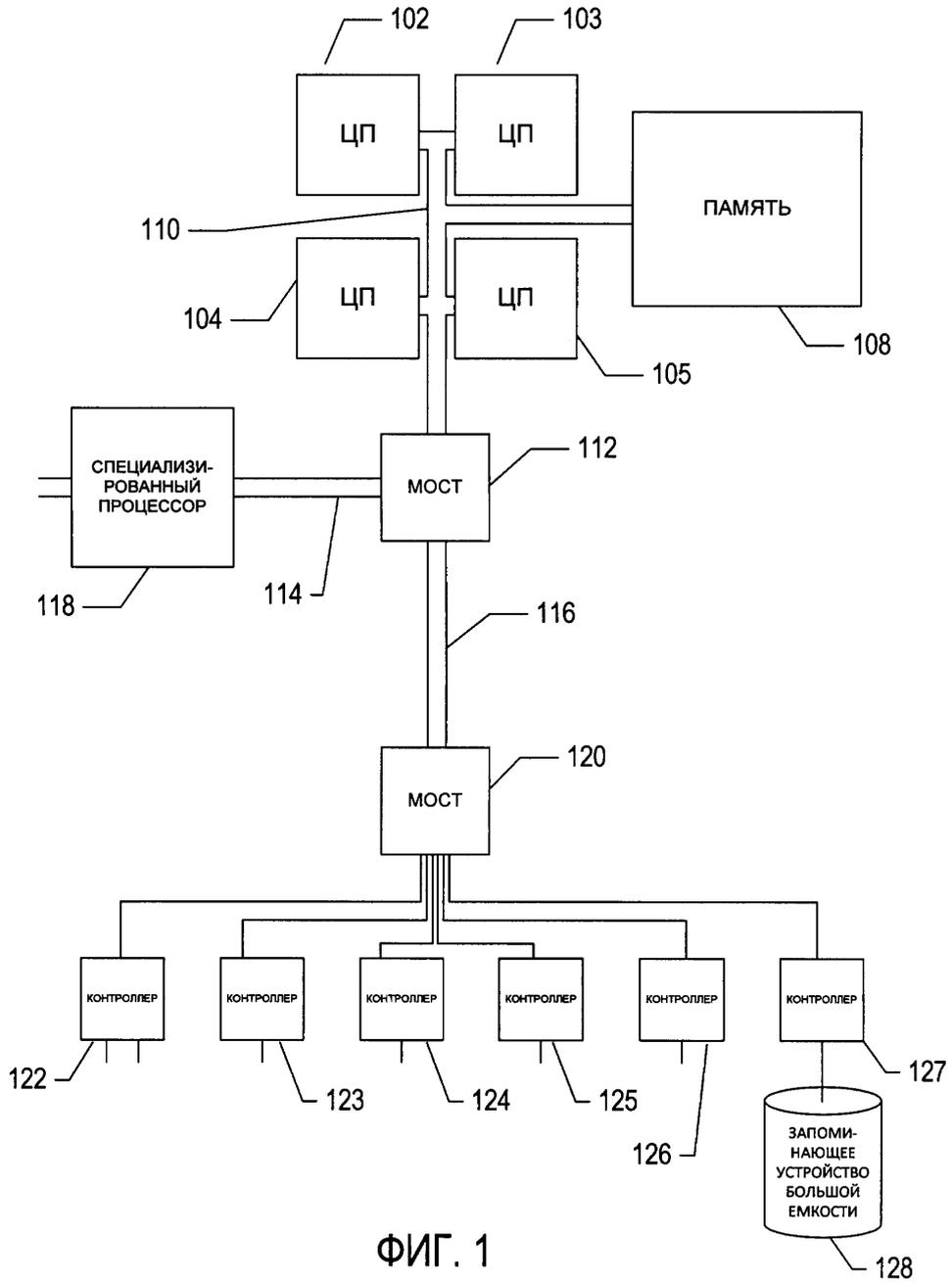
35 страницы, в котором текстовые строки выпрямлены и в котором текстовые знаки и символы имеют прямоугольное расположение.

40

45

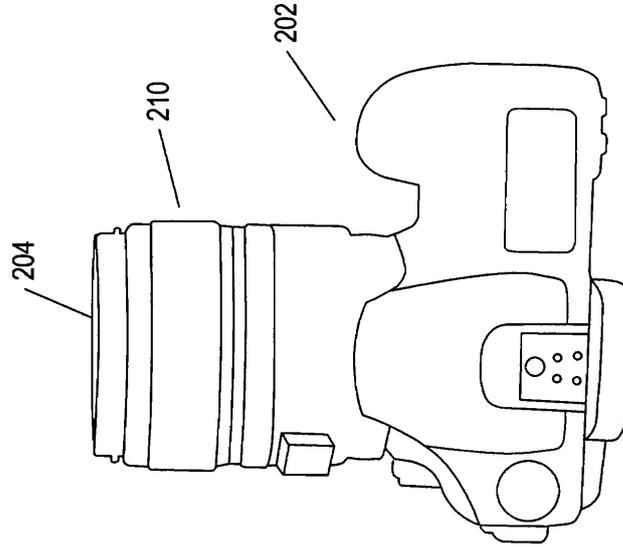
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

1/68

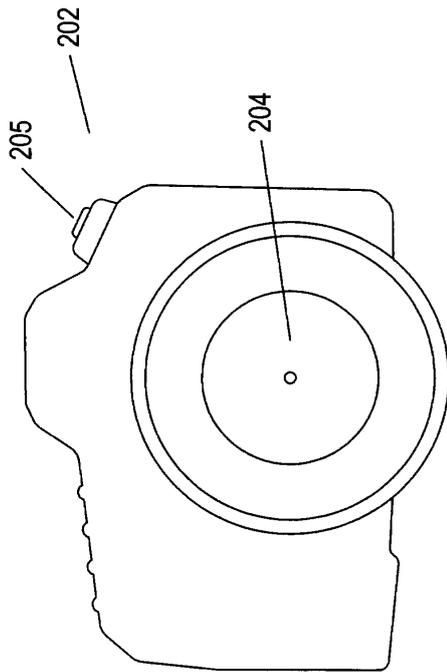


Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

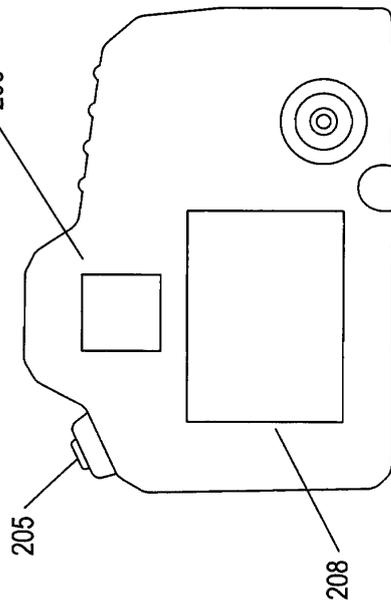
2/68



ФИГ. 2В



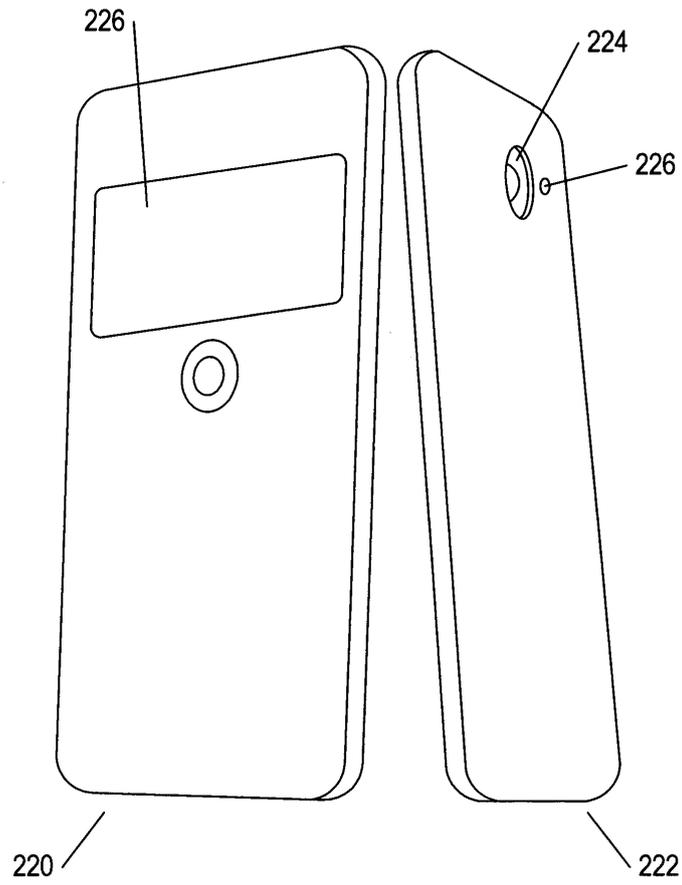
ФИГ. 2А



ФИГ. 2С

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

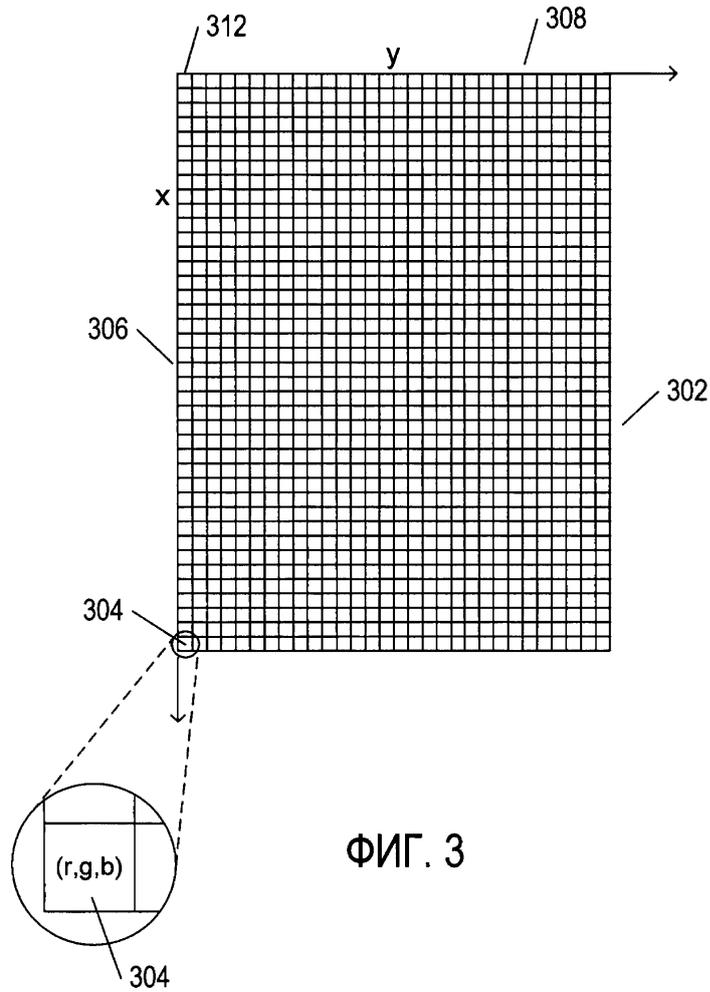
3/68



ФИГ. 2D

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

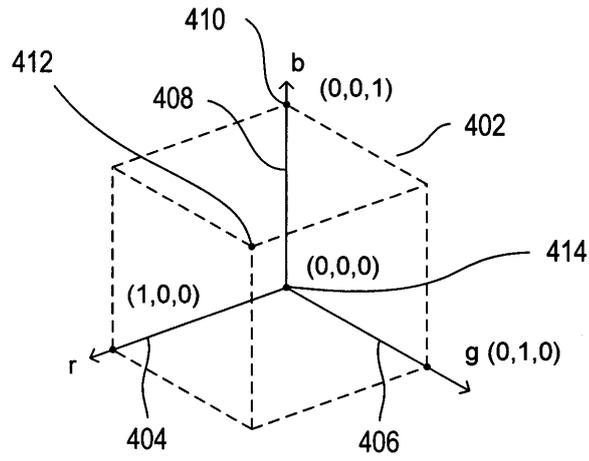
4/68



ФИГ. 3

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

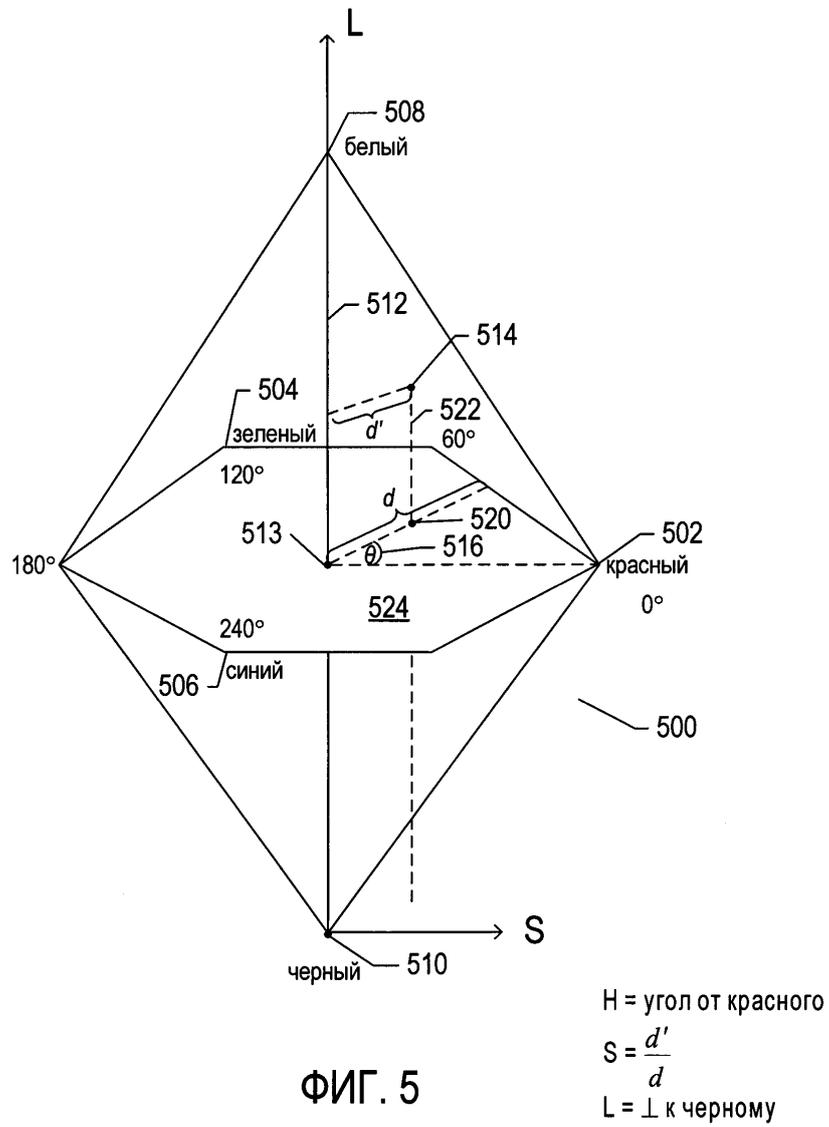
5/68



ФИГ. 4

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-ИИ-0198

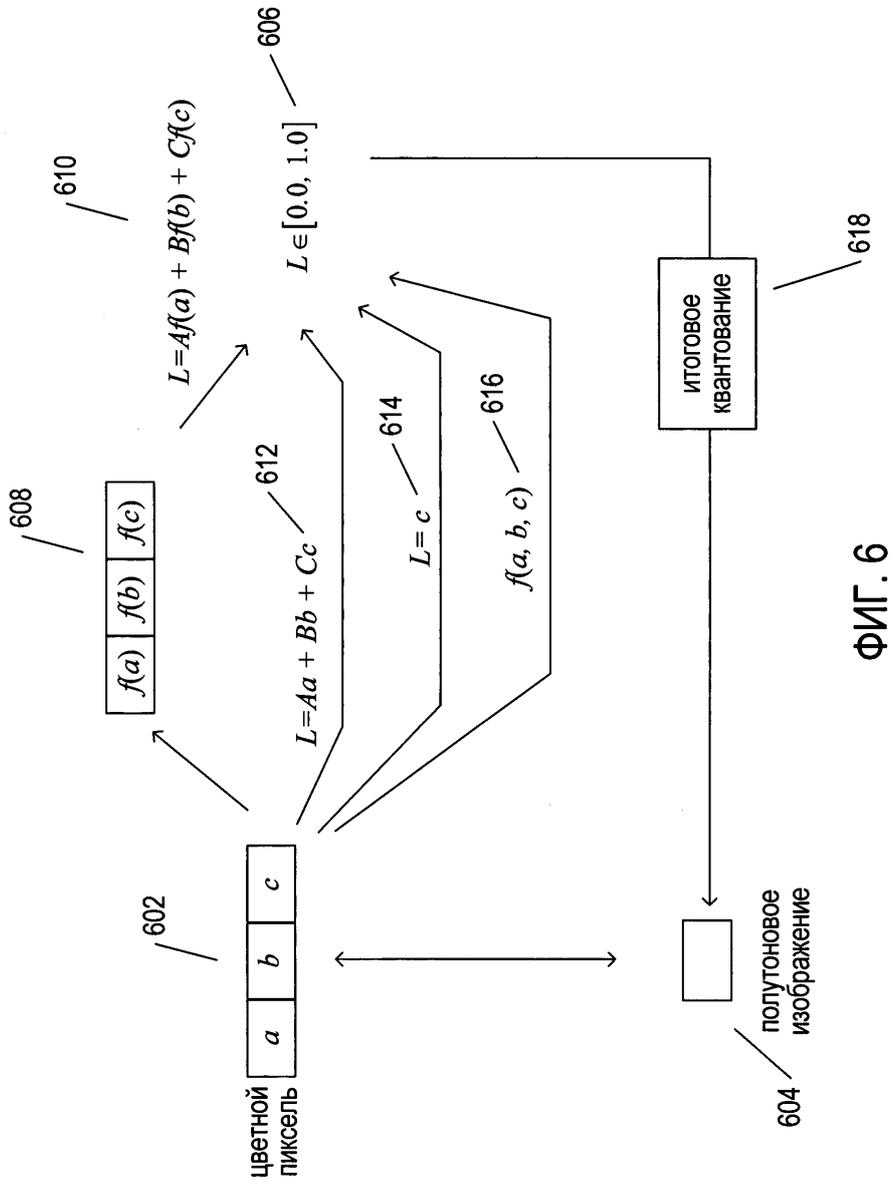
6/68



ФИГ. 5

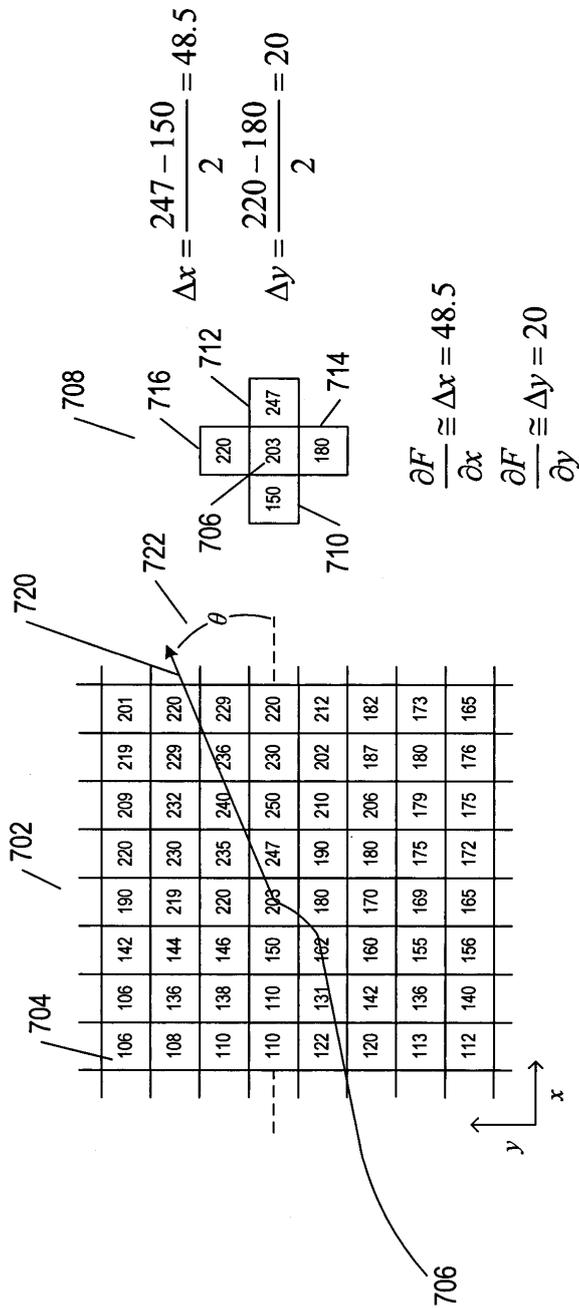
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

7/68



Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУ 2014-IN-0198

8/68



$$\text{gradient} = \nabla F = \frac{\partial F}{\partial x} \mathbf{i} + \frac{\partial F}{\partial y} \mathbf{j} = 48.5 \mathbf{i} + 20 \mathbf{j}$$

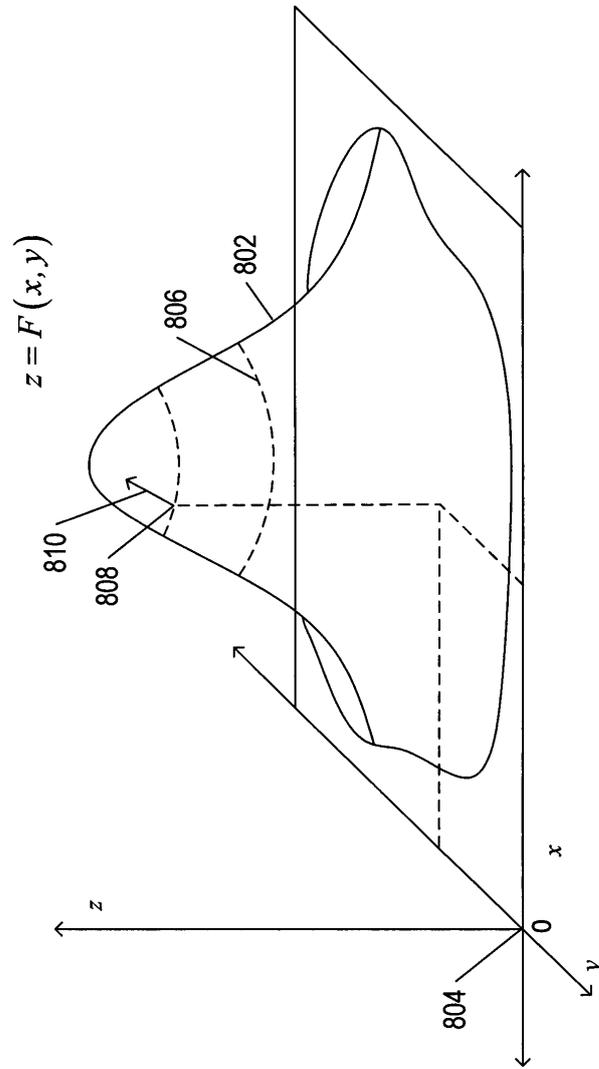
$$|\text{gradient}| = \sqrt{48.5^2 + 20^2} = 52.5$$

$$\theta = \text{atan2}(20, 48.5) = 22.4$$

ФИГ. 7

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

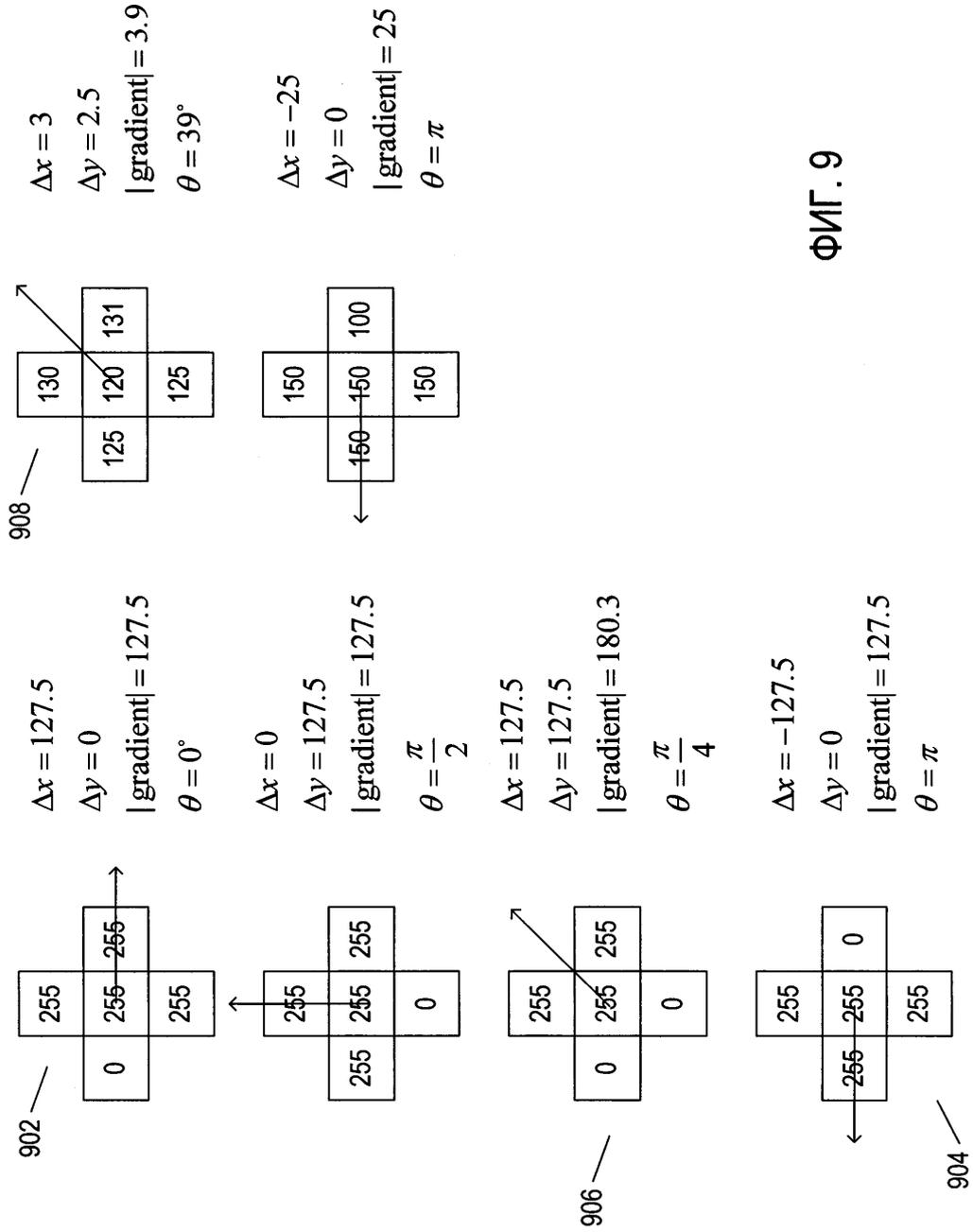
9/68



ФИГ. 8

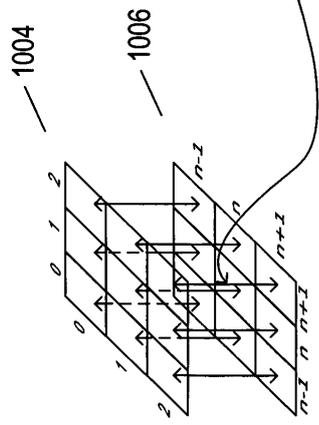
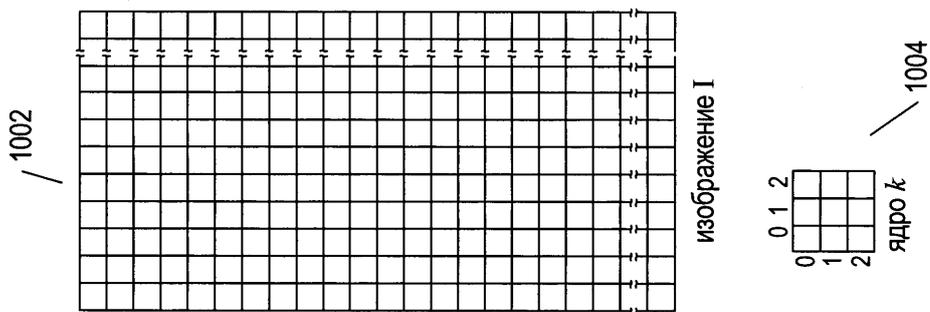
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

10/68



ФИГ. 9

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АБВУУ 2014-IN-0198



11/68

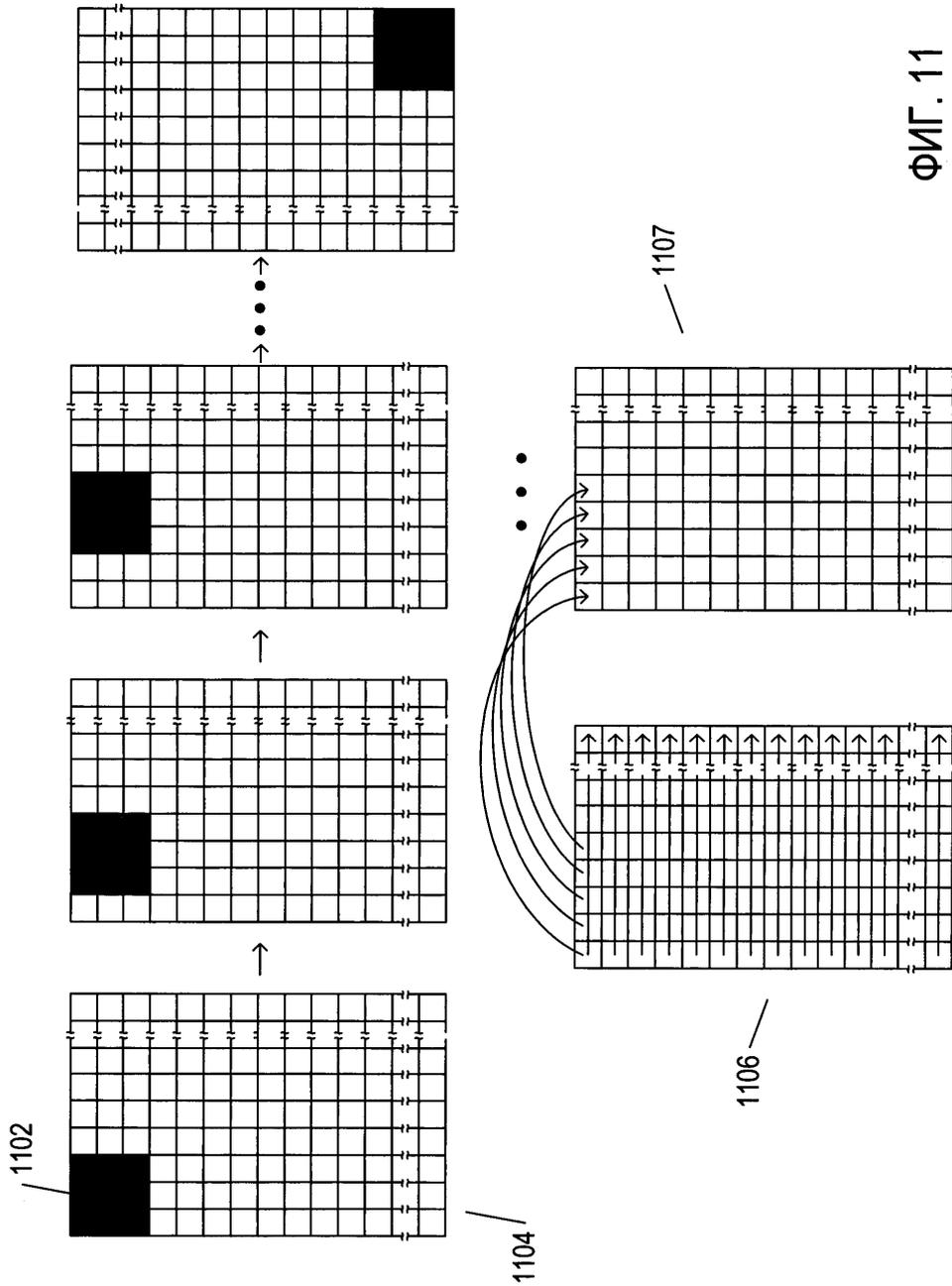
новое значение для  $I_n = k_{0,0} I_{n-1, n-1} + k_{0,1} I_{n-1, n} +$   
 $k_{0,2} I_{n-1, n+1} + k_{1,0} I_{n, n-1} + k_{1,1} I_{n, n} +$   
 $k_{1,2} I_{n, n+1} + k_{2,0} I_{n+1, n-1} + k_{2,1} I_{n+1, n} +$   
 $k_{2,2} I_{n+1, n+1}$  1008

новое значение для  $I_n = f(\text{окрестность } (I_n), k)$  новое 1010  
 значение для  $I_n = f(\text{окрестность } (I_n))$  1012

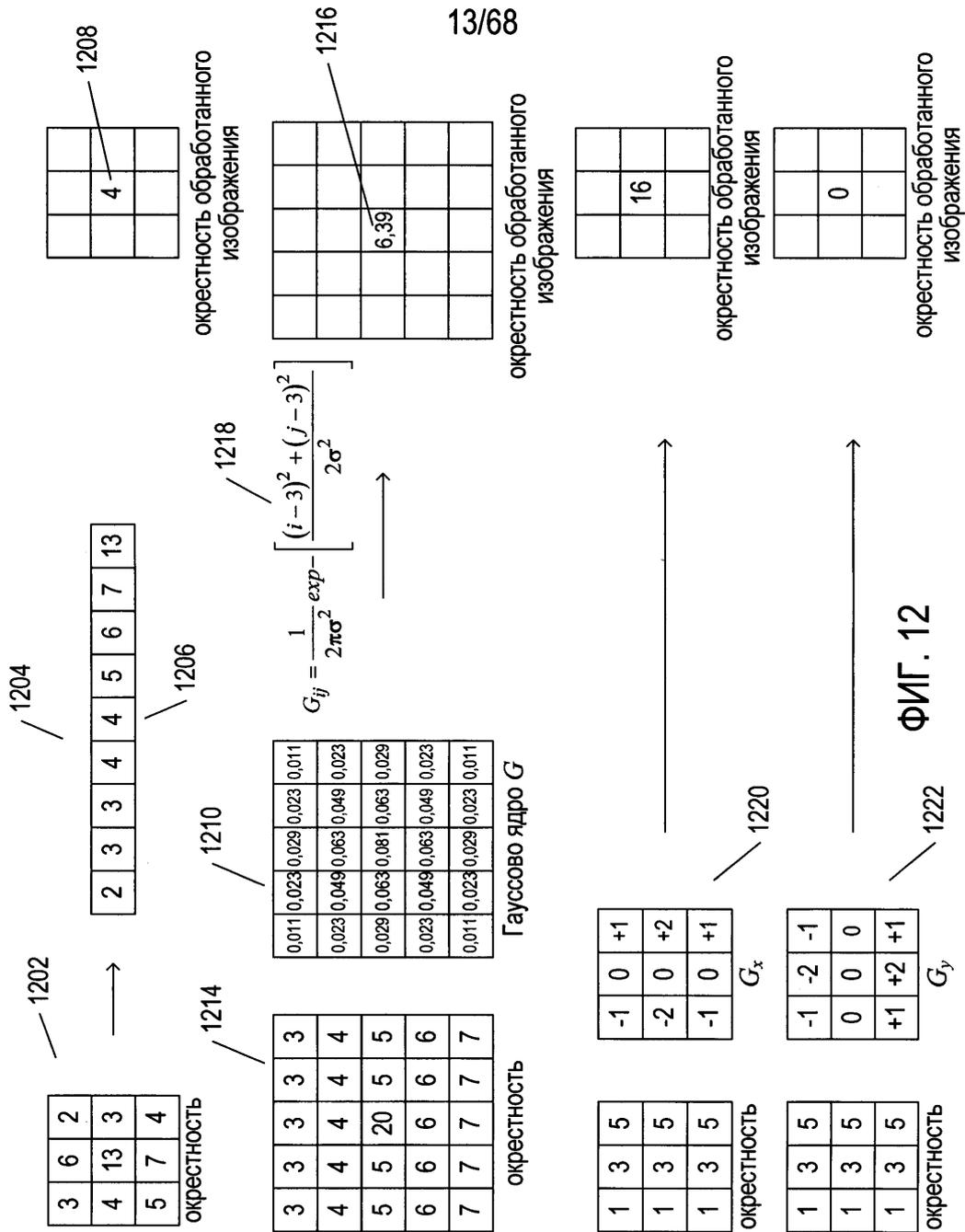
ФИГ. 10

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

12/68

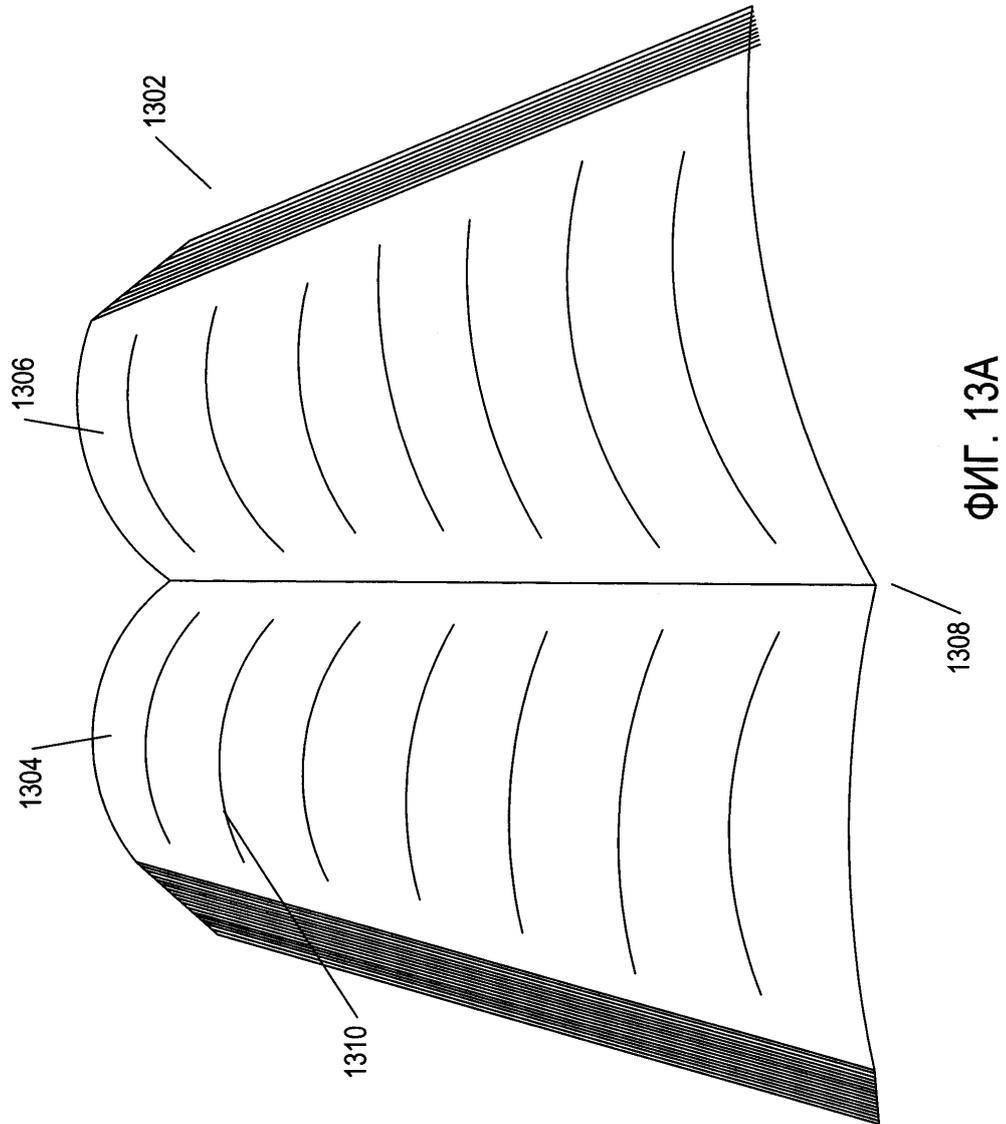


Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198



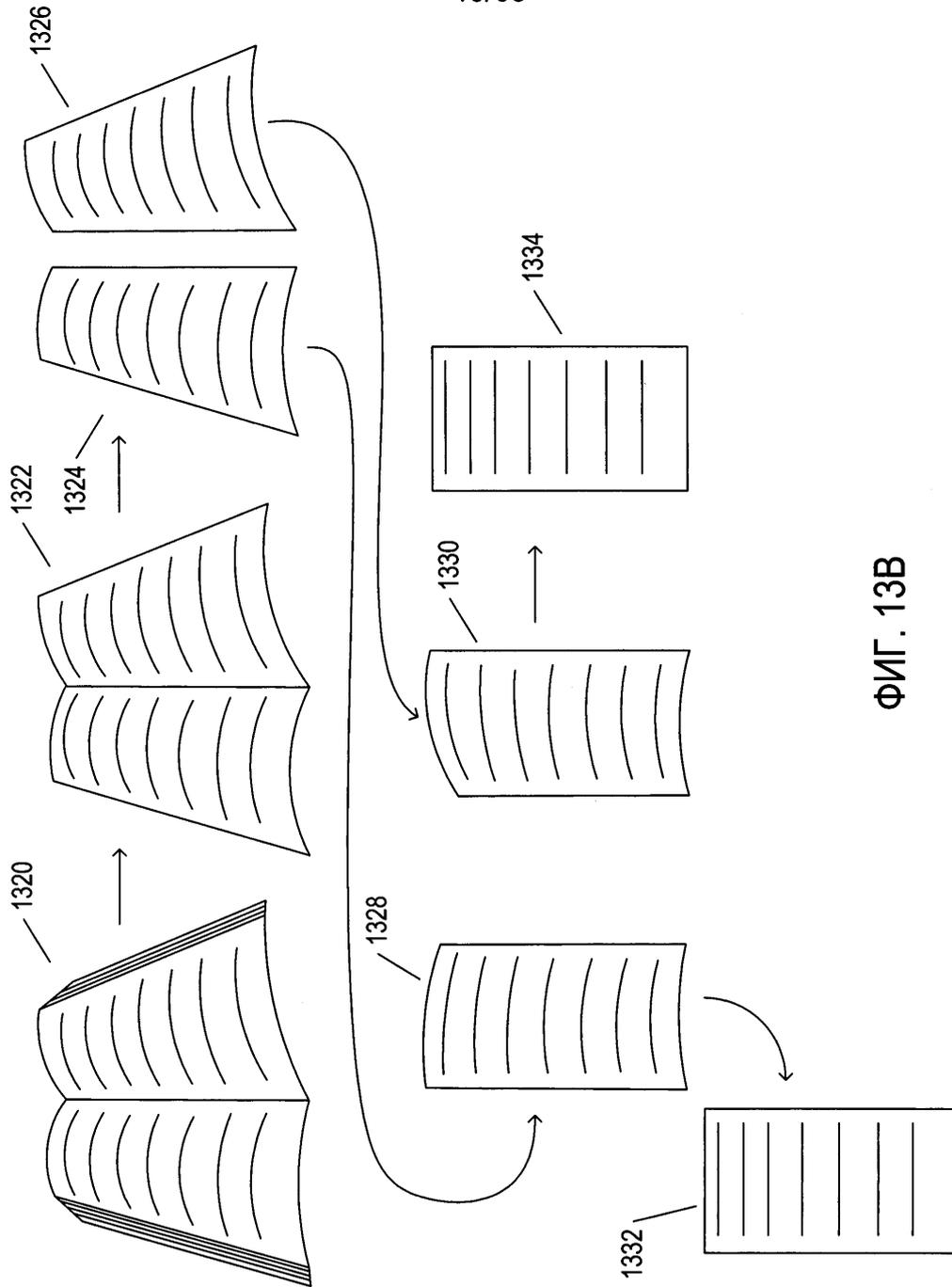
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУ 2014-IN-0198

14/68



Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

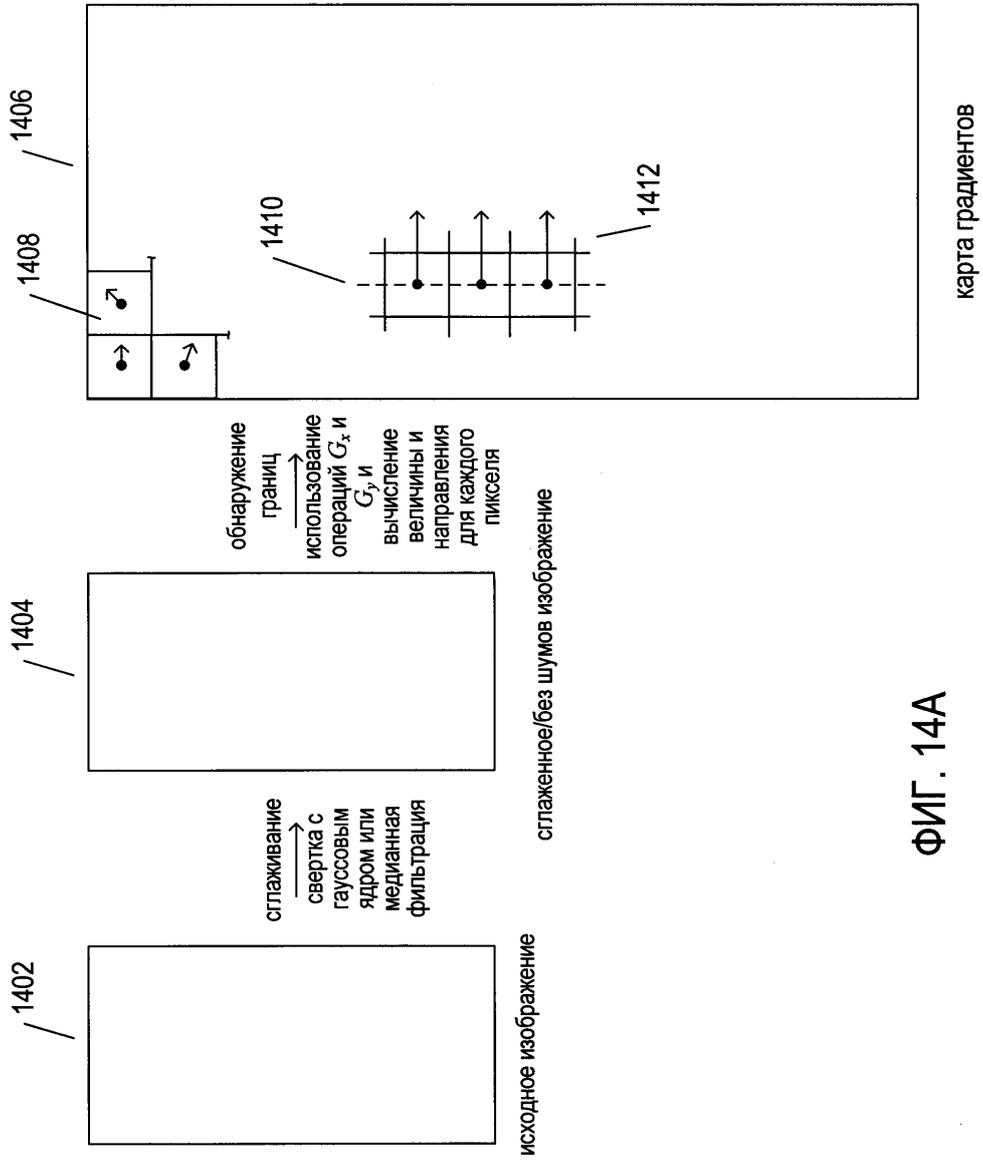
15/68



ФИГ. 13В

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУ 2014-IN-0198

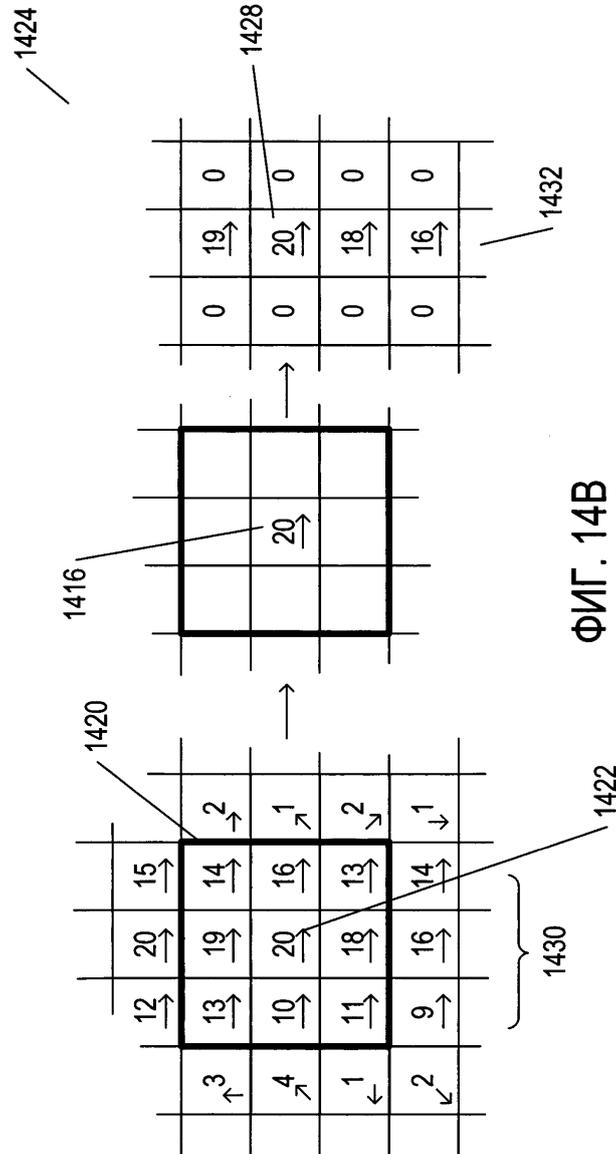
16/68



ФИГ. 14А

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

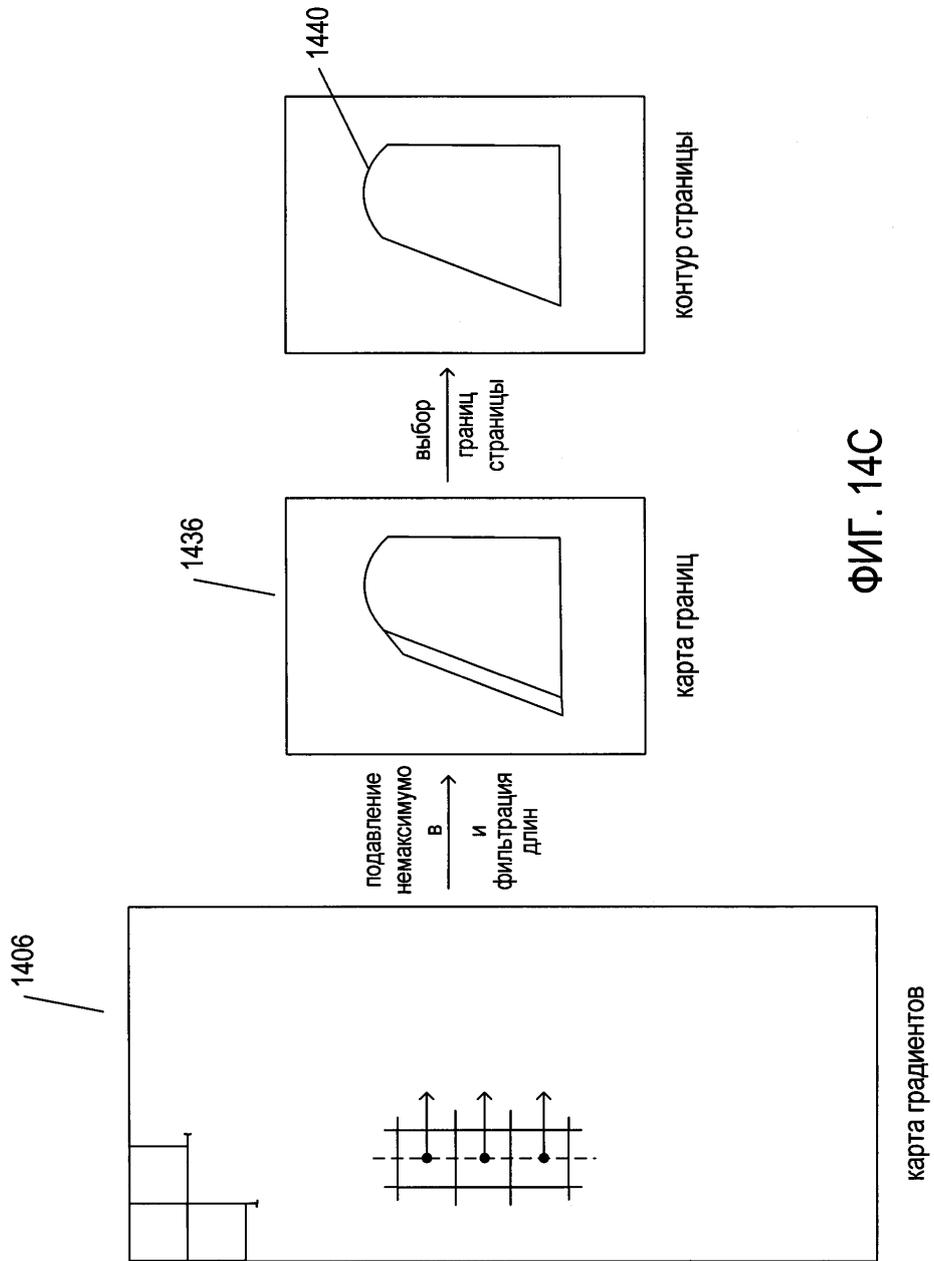
17/68



ФИГ. 14В

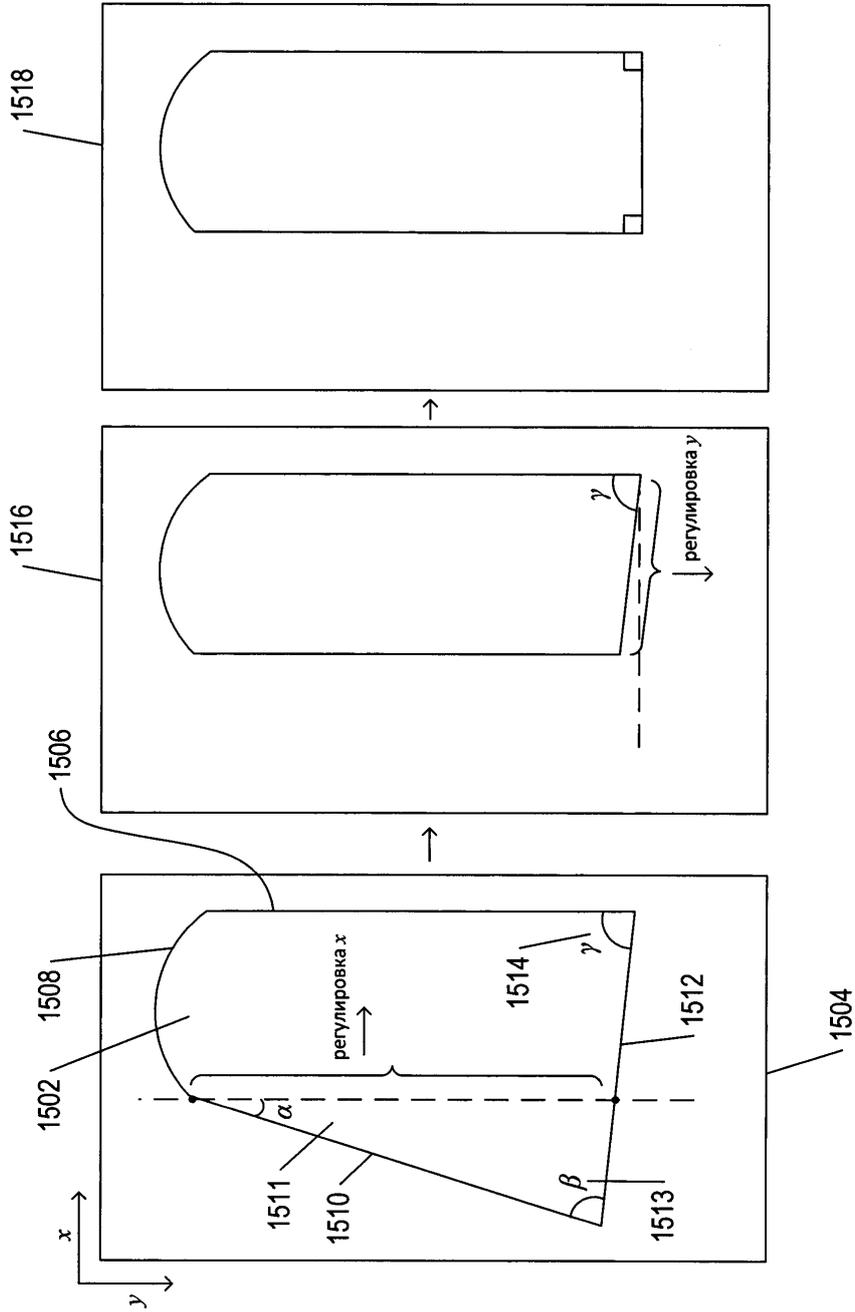
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

18/68



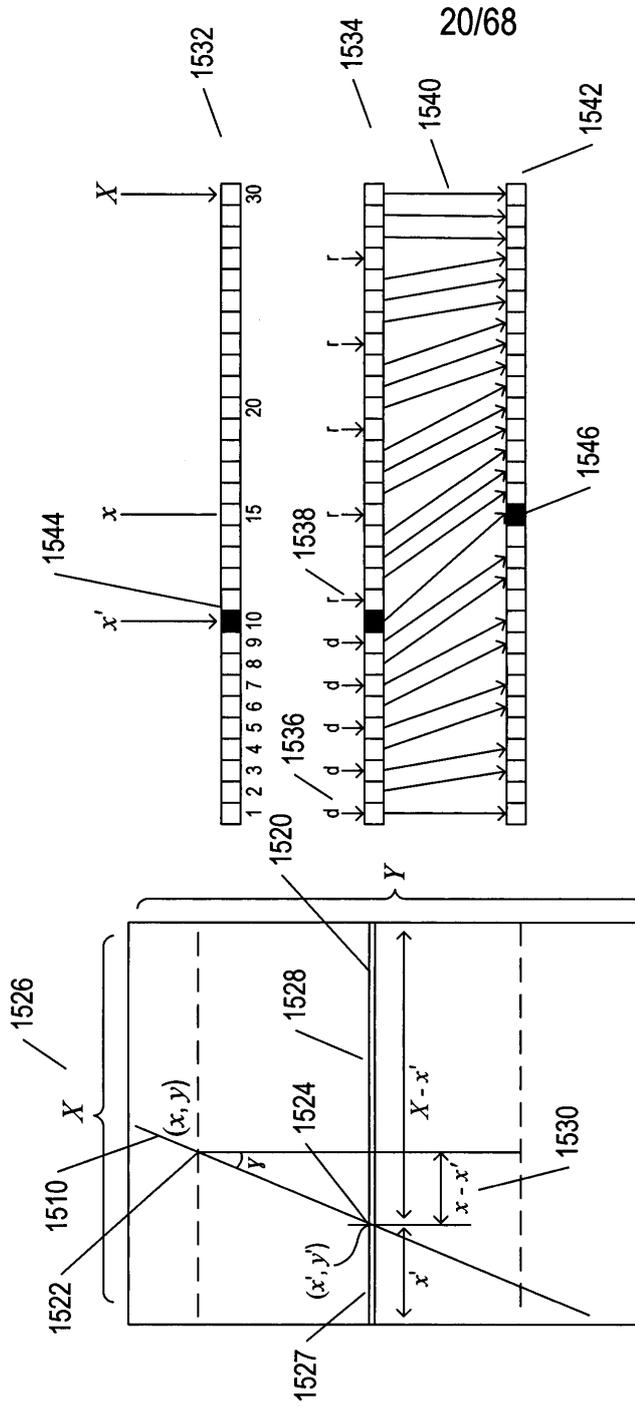
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АБВУУ 2014-IN-0198

19/68



ФИГ. 15А

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198



ФИГ. 15В

удалить  $R = x - x'$  пикселей из правой части

добавить  $R$  пикселей в левую часть

удалить каждый  $\frac{X - x'}{R}$  пиксель, начиная с  $x' + 1$

дублировать каждый  $\frac{x'}{R}$  пиксель, начиная с 1, до  $x' - 1$

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУ 2014-IN-0198

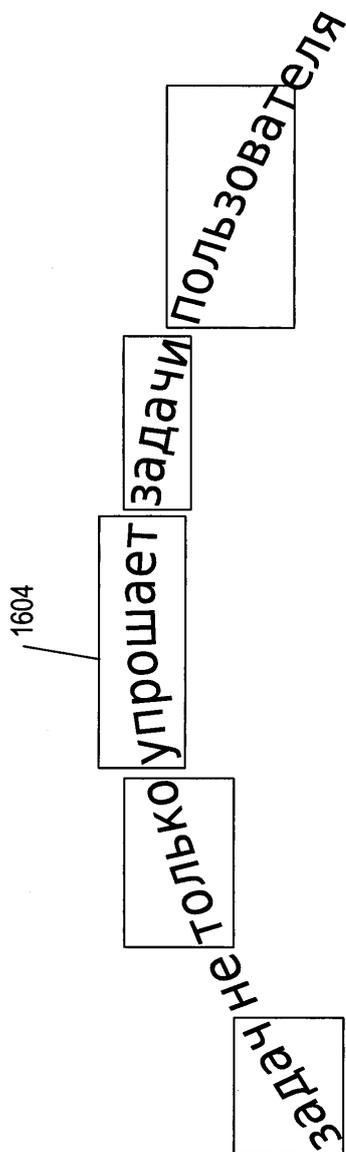
21/68

1602  
задача не только упрощает задачи пользователя

ФИГ. 16А

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУ 2014-IN-0198

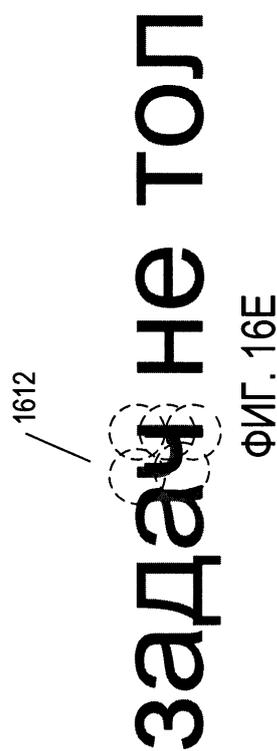
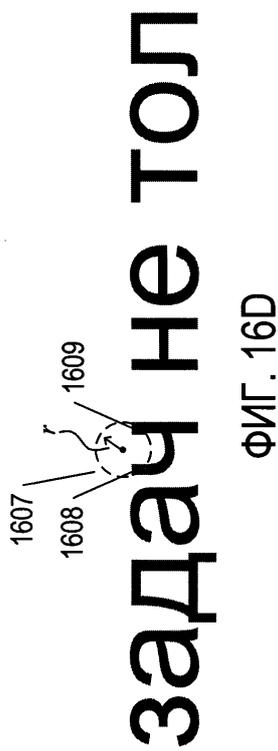
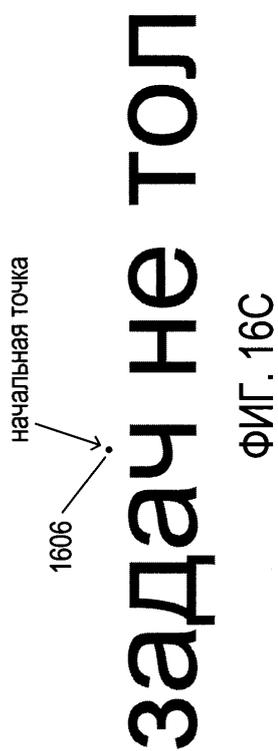
22/68



ФИГ. 16В

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

23/68

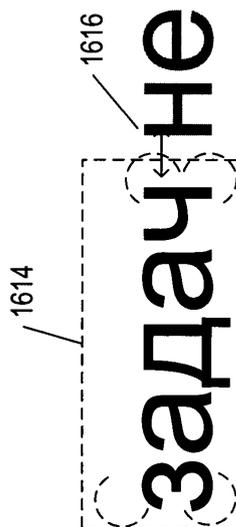


Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

24/68

задач не тол

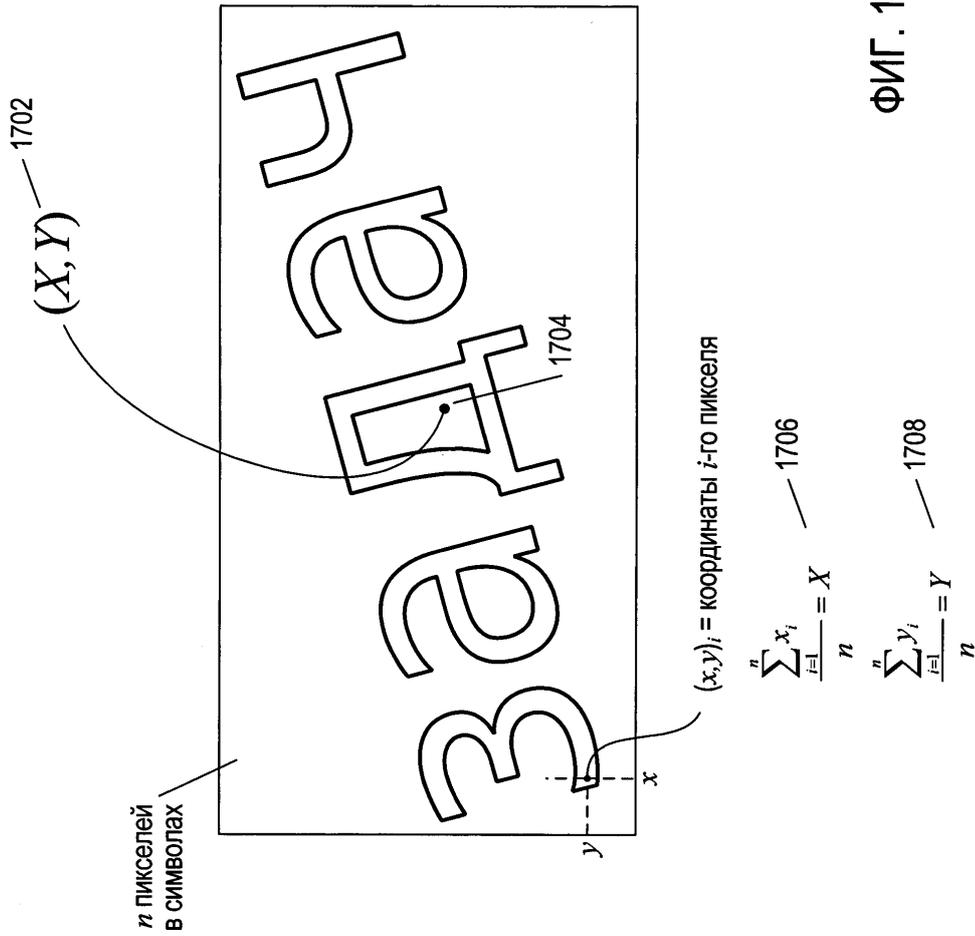
ФИГ. 16F



ФИГ. 16G

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

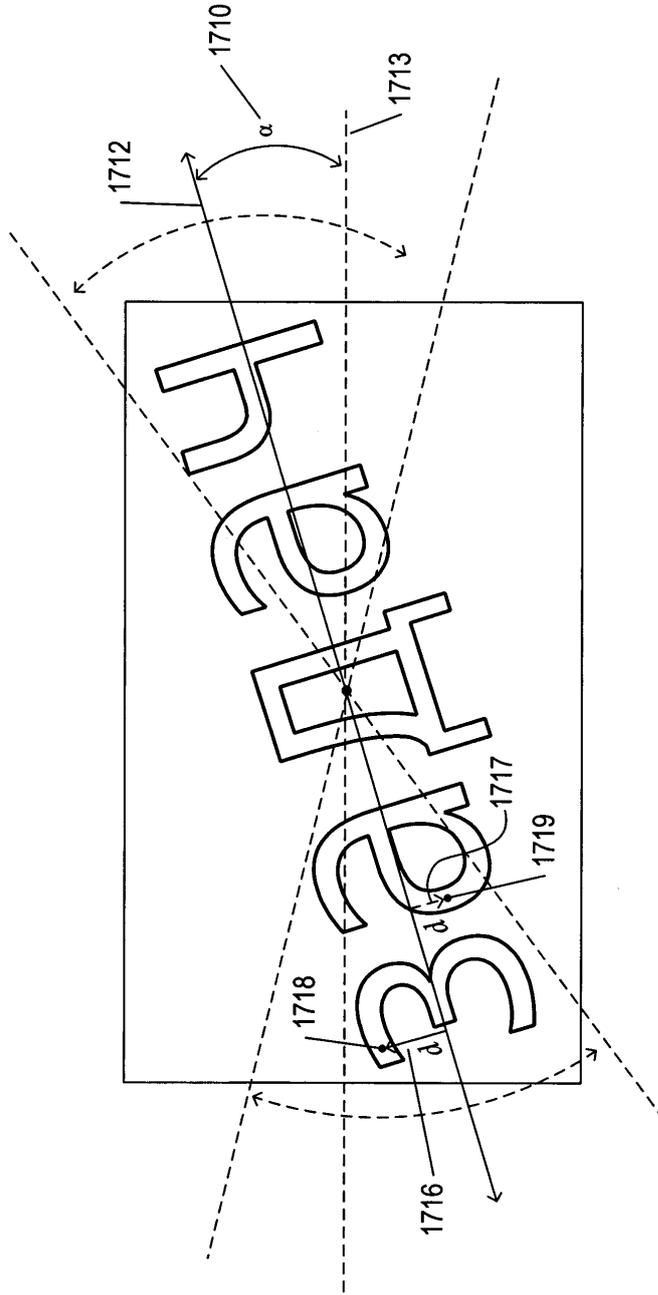
25/68



ФИГ. 17А

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУ 2014-IN-0198

26/68



$$\min_{\alpha} \sum_{i=1}^n d_i$$

1714

ФИГ. 17В

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

1720

$i$	$x$	$y$
1	$x_1$	$y_1$
2	$x_2$	$y_2$
3	$x_3$	$y_3$
4	$x_4$	$y_4$

$n$   $x_n$   $y_n$   
координаты  
пикселя  
символа

1722

$i$	$x^0$	$y^0$
1	$x_1^0$	$y_1^0$
2	$x_2^0$	$y_2^0$
3	$x_3^0$	$y_3^0$
4	$x_4^0$	$y_4^0$

$n$   $x_n^0$   $y_n^0$   
преобразованные  
координаты пикселя  
символа

1724  $K_1 = \sum_{i=1}^n (x_i^0)^2 - (y_i^0)^2$

1725  $K_2 = \sum_{i=1}^n x_i^0 y_i^0$

1726  $K = \frac{K_1}{K_2}$

1730  $\tan \alpha = \frac{-K \pm \sqrt{K^2 + 4}}{2}$

1728 берем  $\alpha$  как  $\alpha$ , расположенный ближе к горизонтали (0)

ФИГ. 17С

1732

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad 1734$$

1736  $H = \sum_{i=1}^n (y_i^{0*})^2$

1738  $y_i^{0*} = -x_i^0 \sin \alpha + y_i^0 \cos \alpha$

1740  $\frac{d(y_i^{0*})^2}{d\alpha} = 2y_i^{0*} \frac{dy_i^{0*}}{d\alpha}$

1742  $= 2[-x_i^0 \sin \alpha + y_i^0 \cos \alpha][ -x_i^0 \cos \alpha - y_i^0 \sin \alpha ]$

1744  $\frac{dH}{d\alpha} = \sum_{i=1}^n 2[-x_i^0 \sin \alpha + y_i^0 \cos \alpha][ -x_i^0 \cos \alpha - y_i^0 \sin \alpha ]$

1746  $\min \sum_{i=1}^n d_i$  когда  $\frac{dH}{d\alpha} = 0$

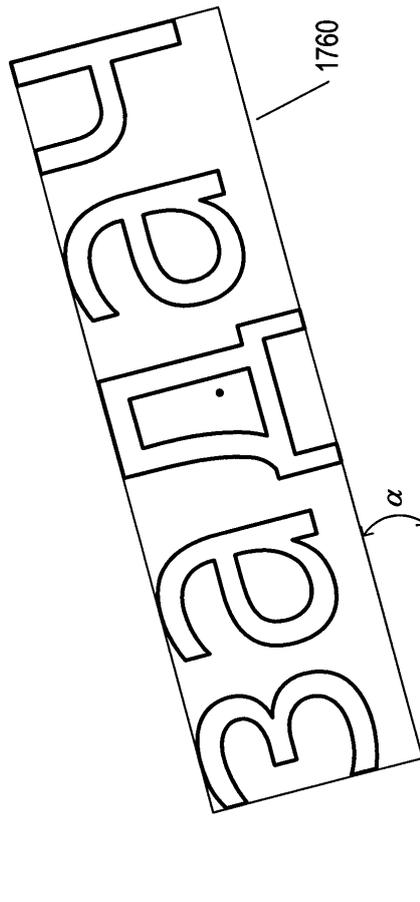
1748  $0 = \left( \sum_{i=1}^n x_i^0 y_i^0 \right) \sin^2 \alpha + \left( \sum_{i=1}^n (x_i^0)^2 - (y_i^0)^2 \right) \sin \alpha \cos \alpha - \left( \sum_{i=1}^n x_i^0 y_i^0 \right) \cos^2 \alpha$

1750  $\tan^2 \alpha + K \tan(\alpha) - 1 = 0,$

где  $K = \left( \sum_{i=1}^n (x_i^0)^2 - (y_i^0)^2 \right) / \sum_{i=1}^n x_i^0 y_i^0$

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУ 2014-IN-0198

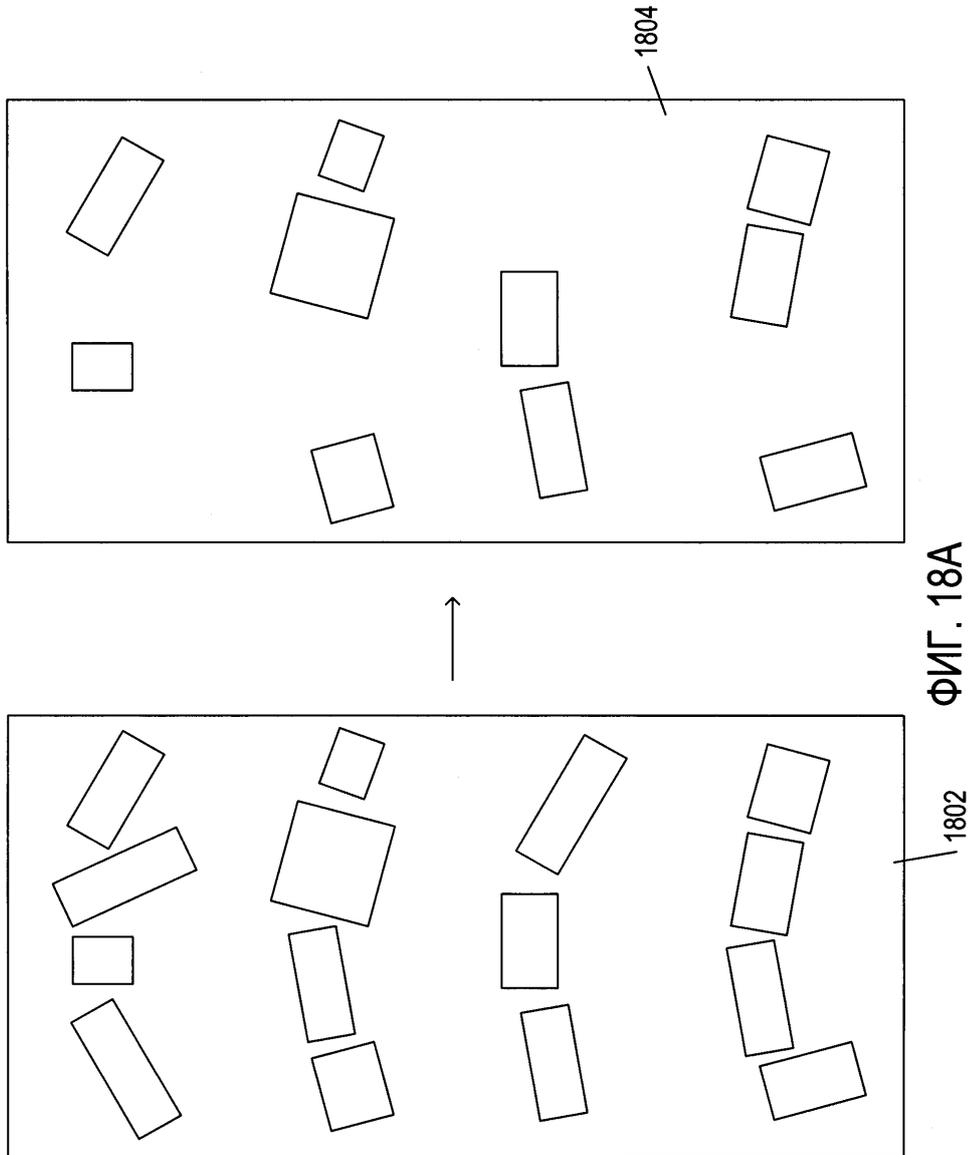
28/68



ФИГ. 17D

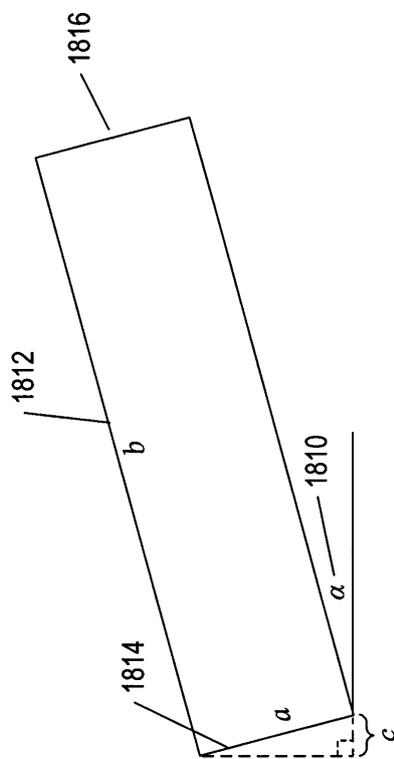
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

29/68



Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

30/68



Отклоняем, когда  $\frac{a}{c} \leq T_1$ ;

отклоняем, когда длина  $a$ , в пикселях,  $< T_2$ ; и

отклоняем, когда  $\frac{a}{H}$  или  $\frac{b}{W} > T_3$

где  $H$  - высота изображения;

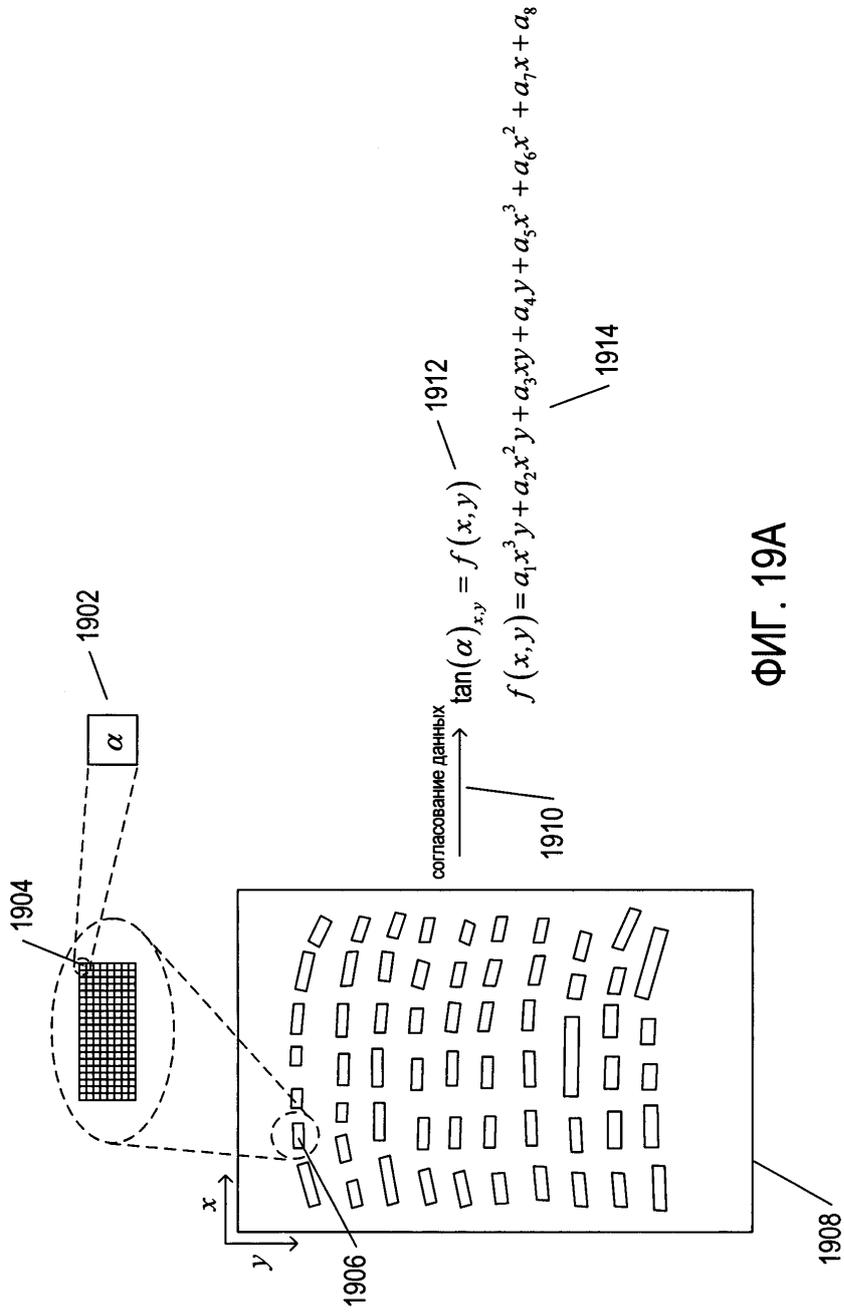
$W$  - ширина изображения; и

$T_1, T_2, \text{ и } T_3$  - пороговые значения.

ФИГ. 18В

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

31/68

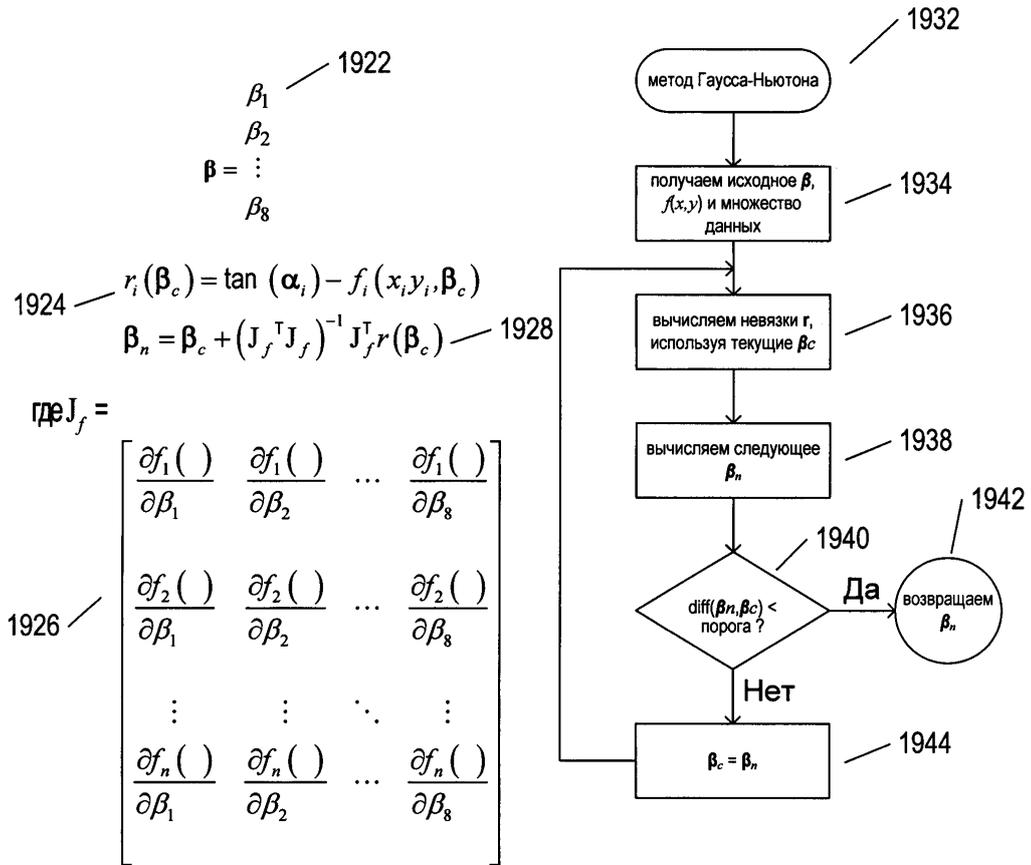


ФИГ. 19А

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

32/68

$f(x, y) = \tan \alpha = \beta_1 x^3 y + \beta_2 x^2 y + \beta_3 xy + \beta_4 y + \beta_5 x^3 + \beta_6 x^2 + \beta_7 x + \beta_8$  — 1920  
необходимо найти  $\beta_0, \dots, \beta_7$ , основанные на множестве значений  $\{\tan \alpha_i, (x, y)_i\}$   
для  $i \in \{1, 2, 3, \dots, n\}$  где  $n$  - количество пикселей

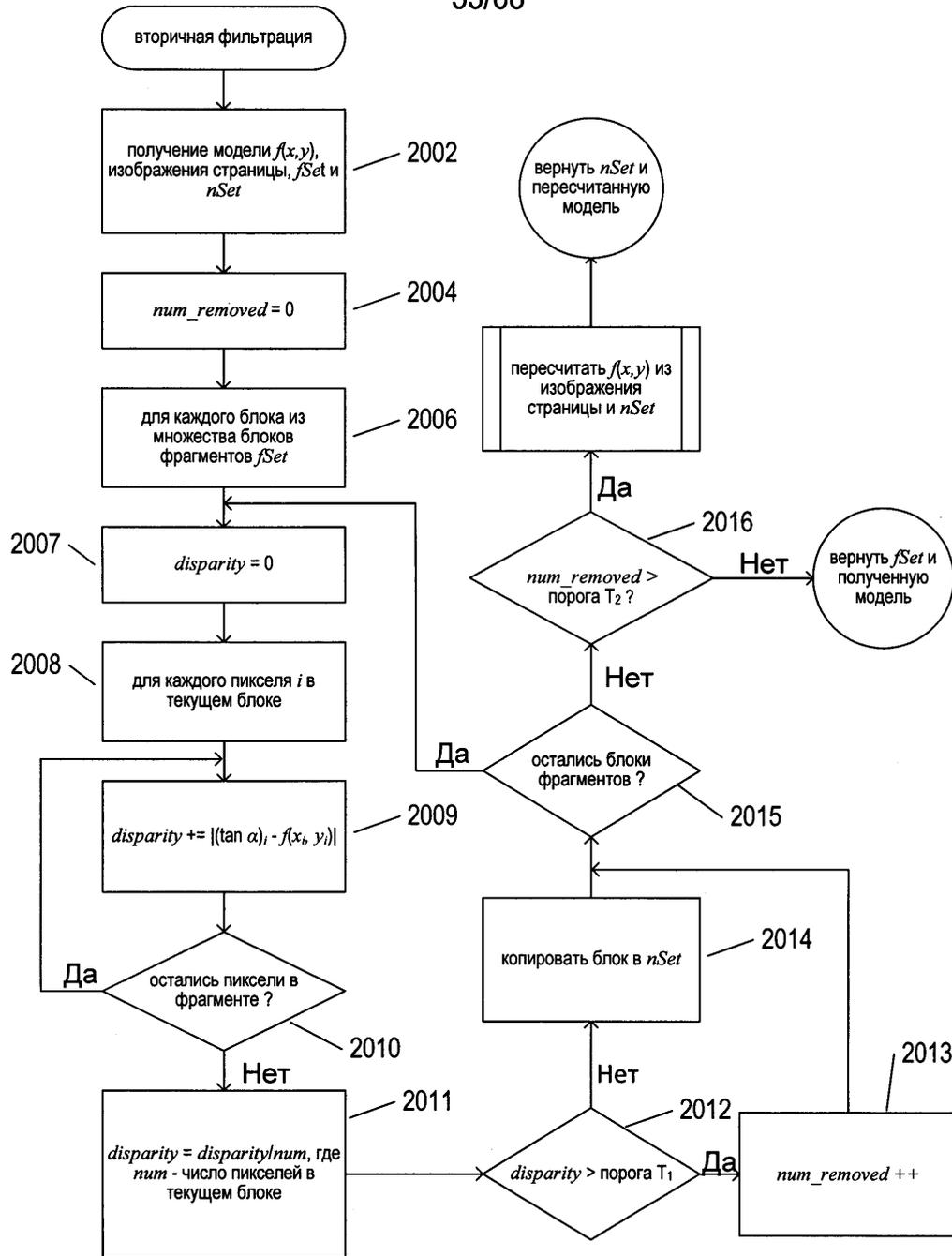


Метод Гаусса-Ньютона минимизирует  $\sum_{i=1}^n r_i^2(\beta)$

1930 /

ФИГ. 19В

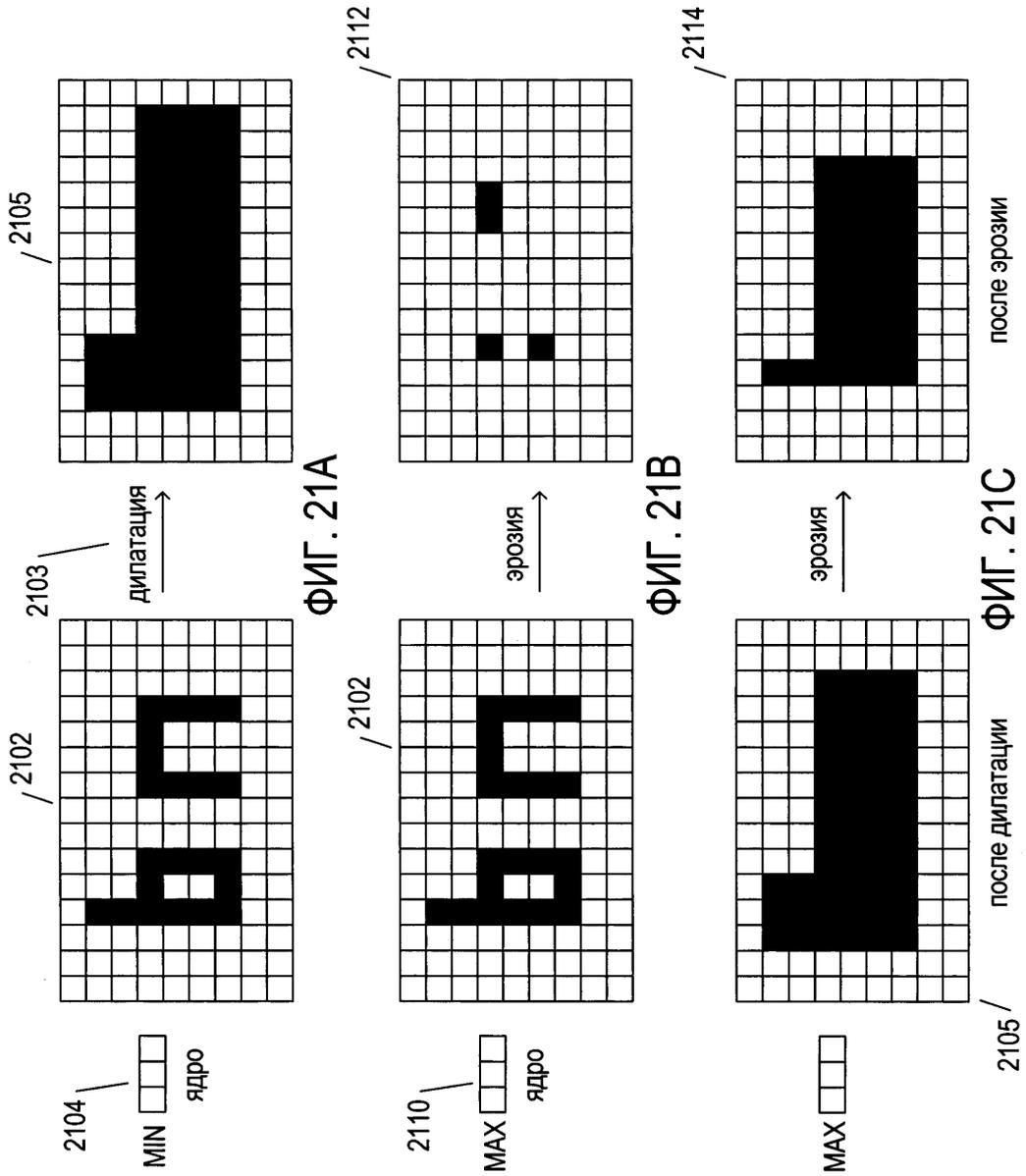
33/68



ФИГ. 20

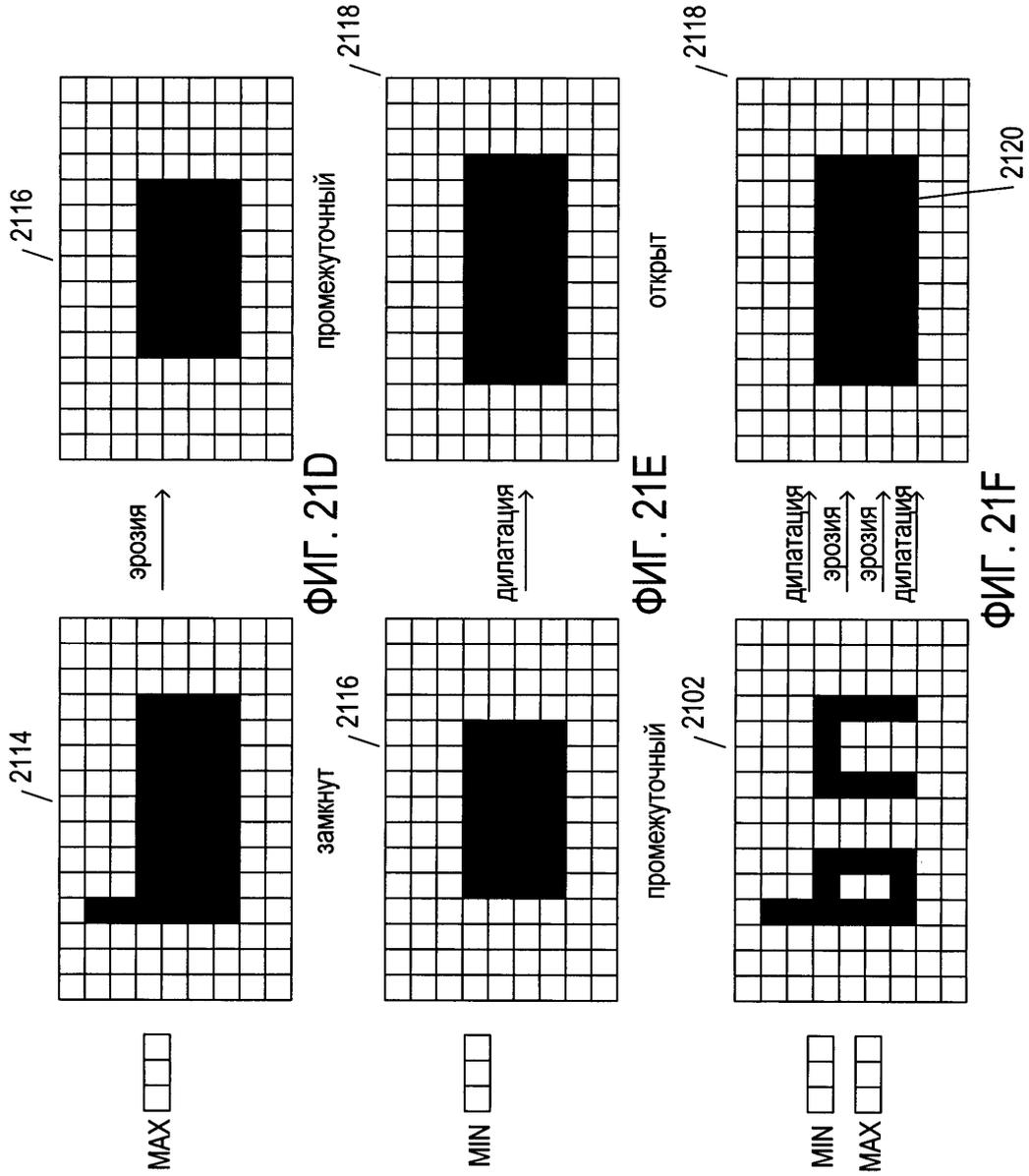
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

34/68



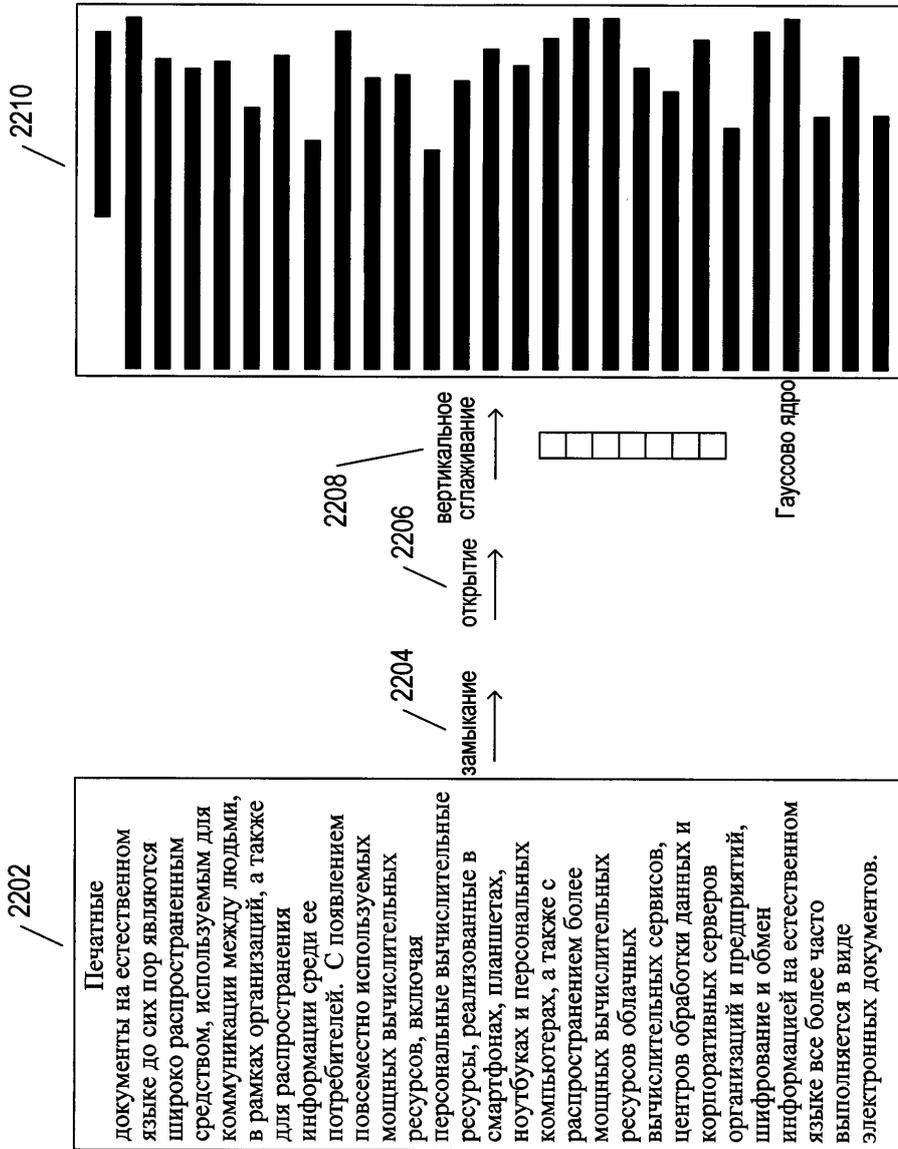
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

35/68



Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

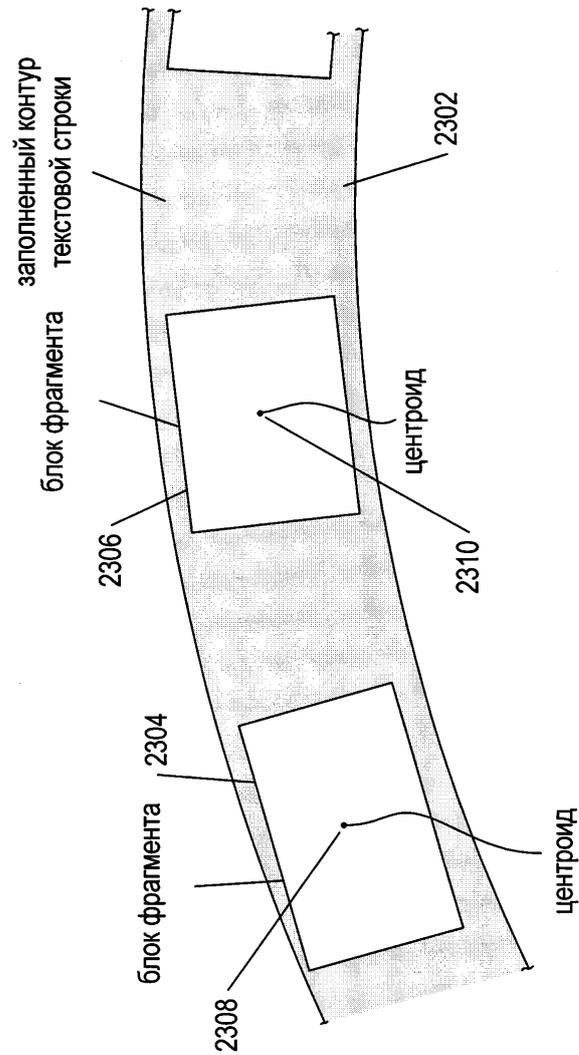
36/68



ФИГ. 22

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

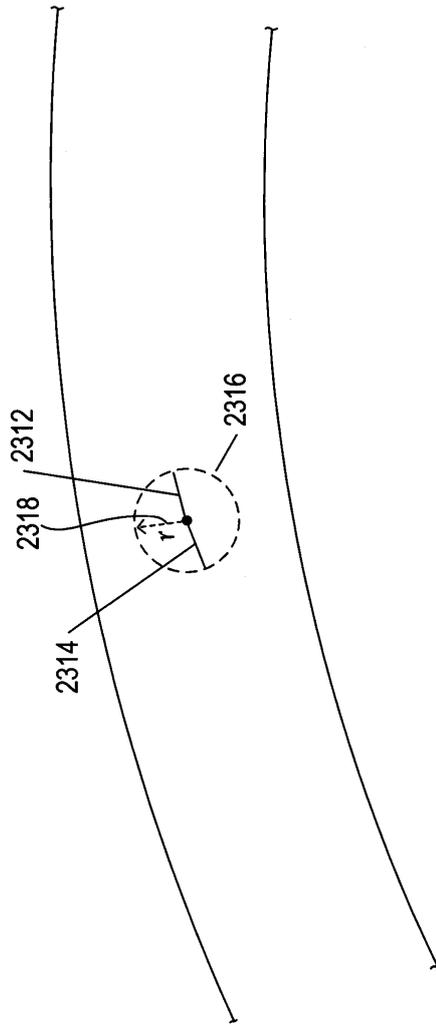
37/68



ФИГ. 23А

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

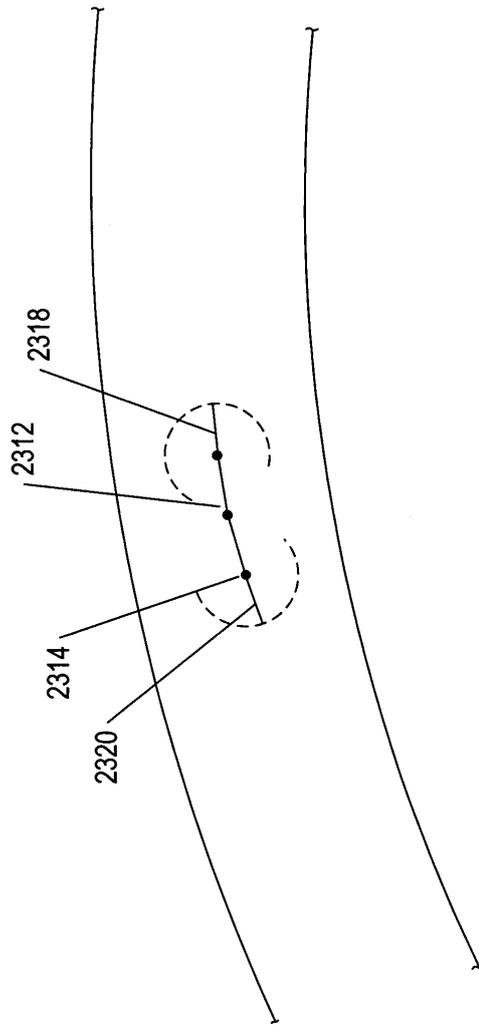
38/68



ФИГ. 23В

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУ 2014-IN-0198

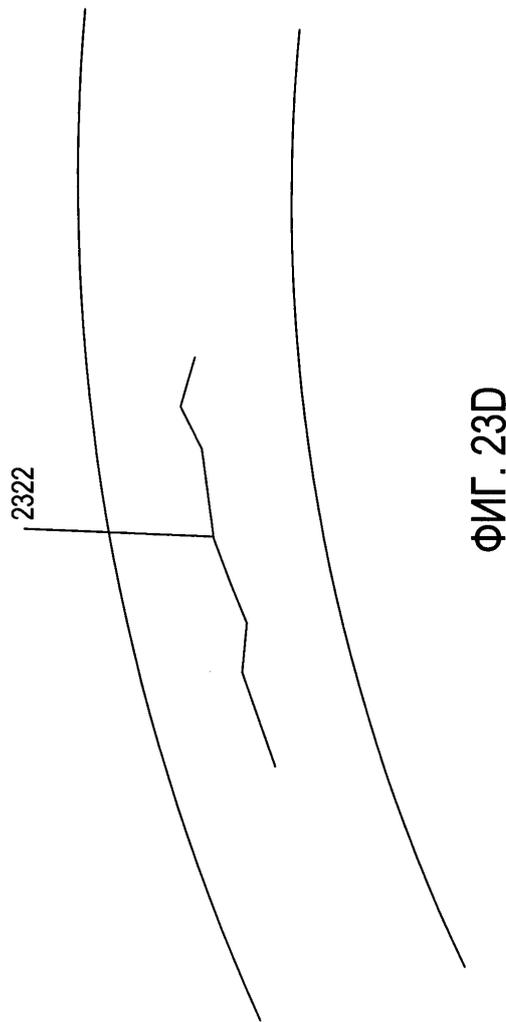
39/68



ФИГ. 23С

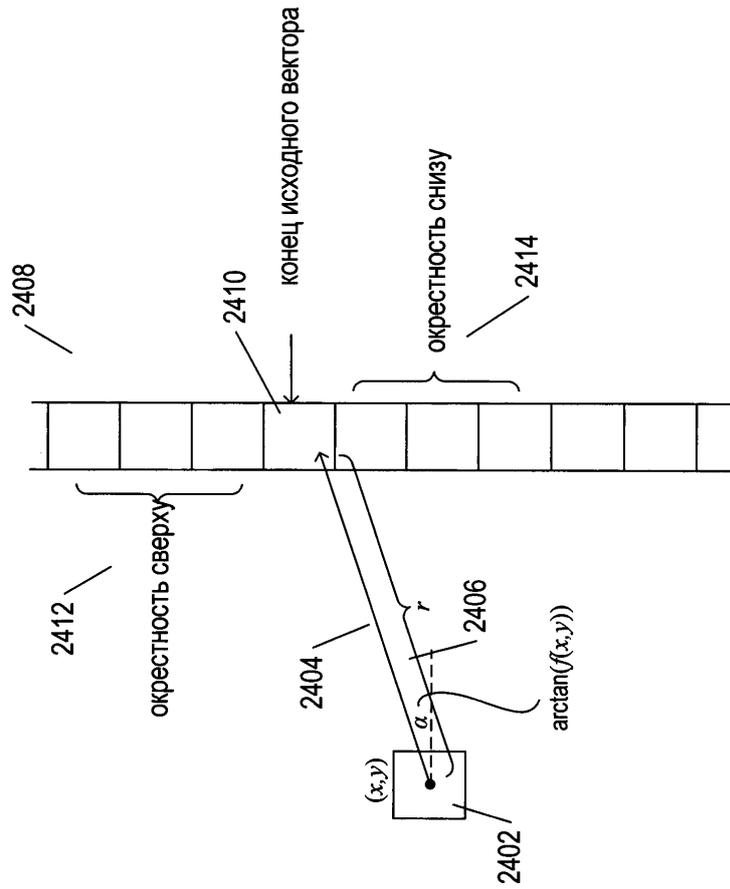
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУ 2014-IN-0198

40/68



Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

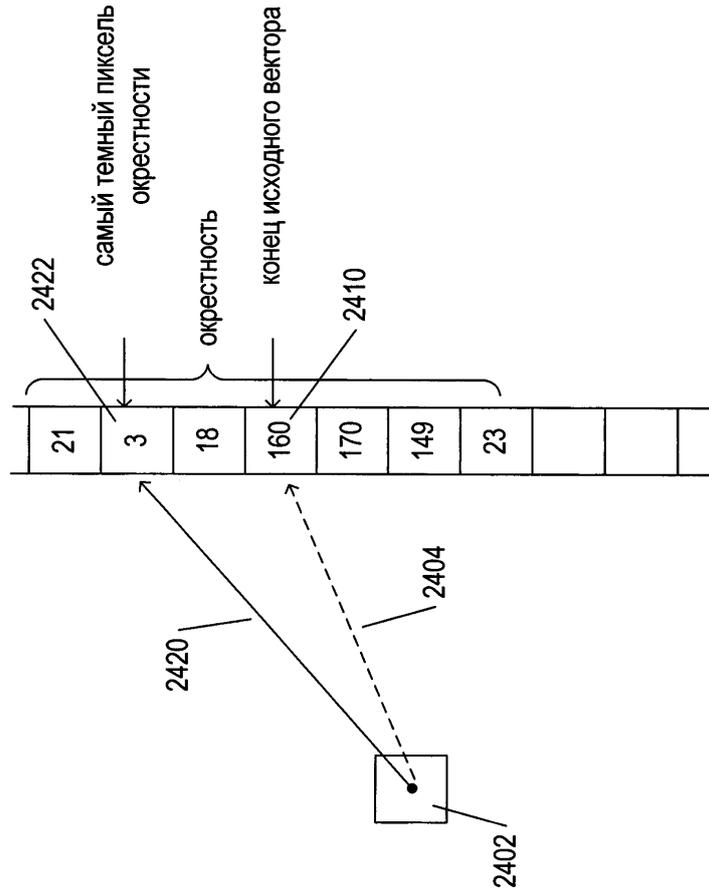
41/68



ФИГ. 24А

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

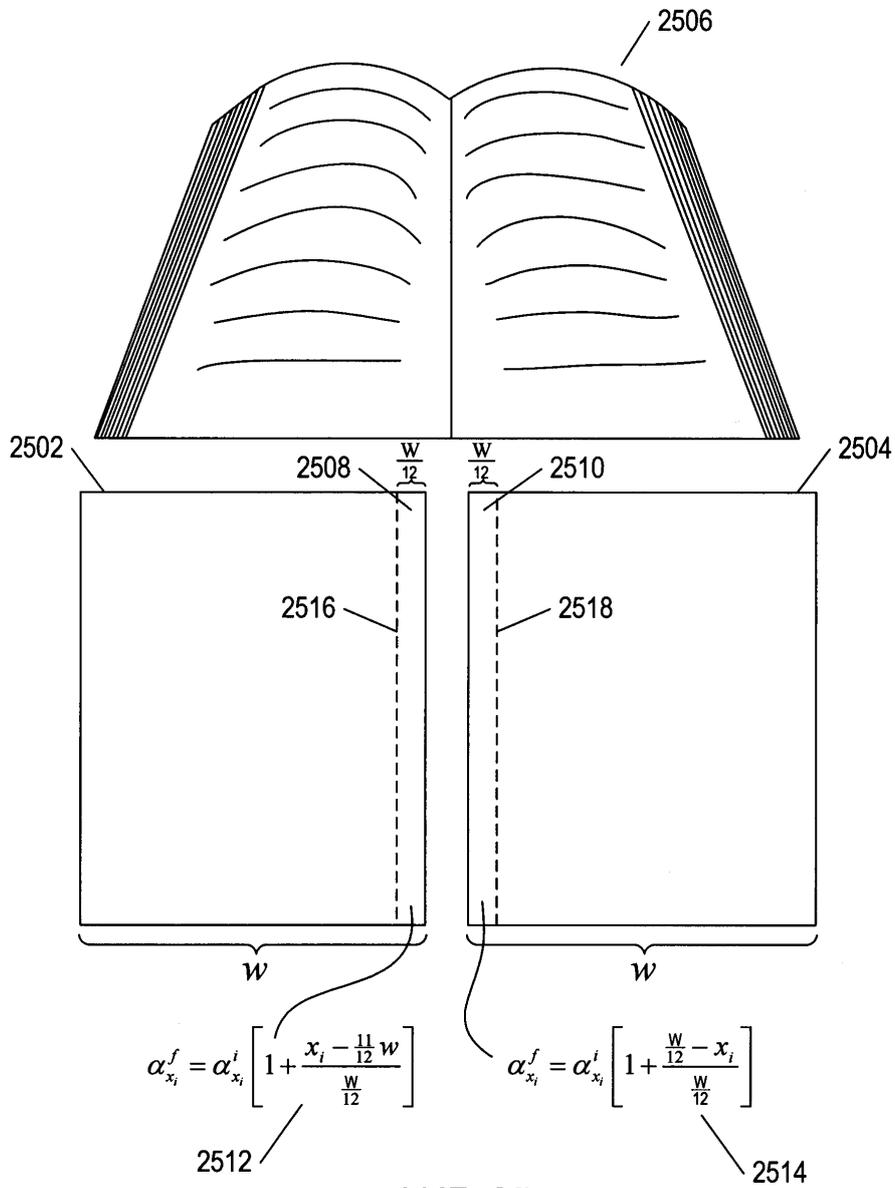
42/68



ФИГ. 24В

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУ 2014-IN-0198

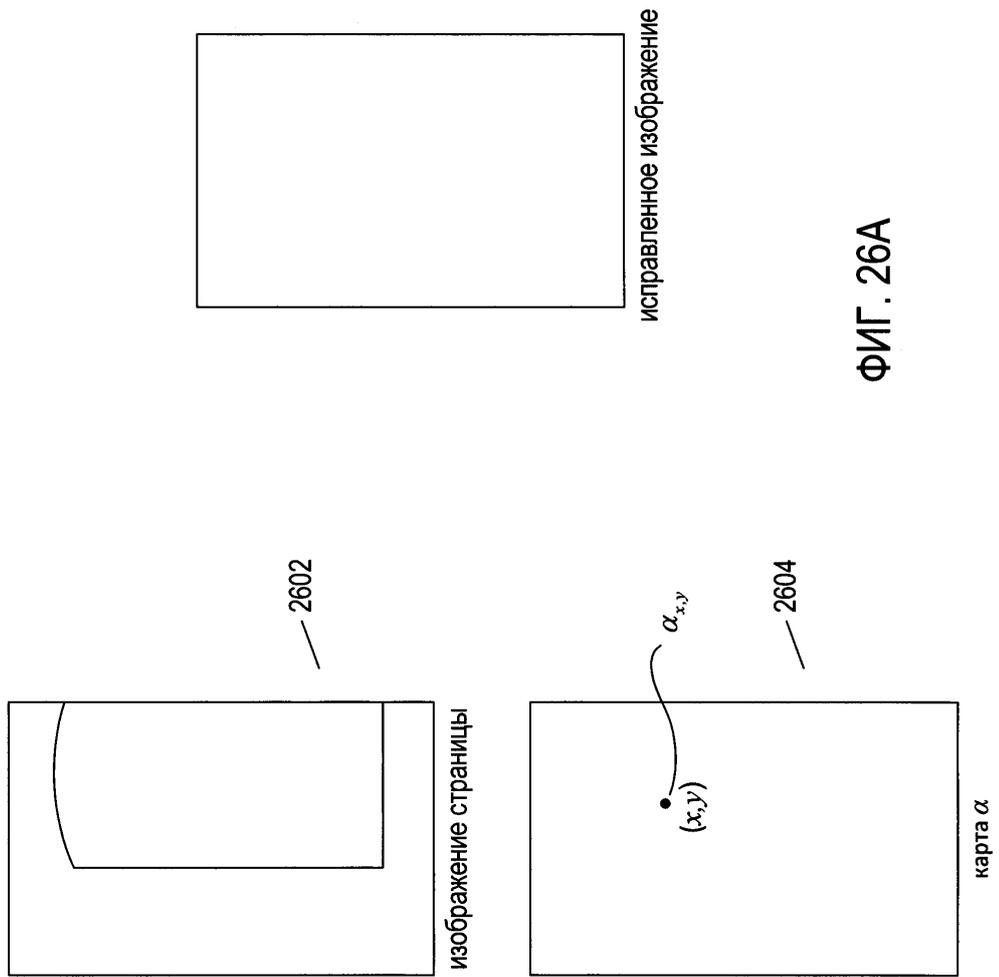
43/68



ФИГ. 25

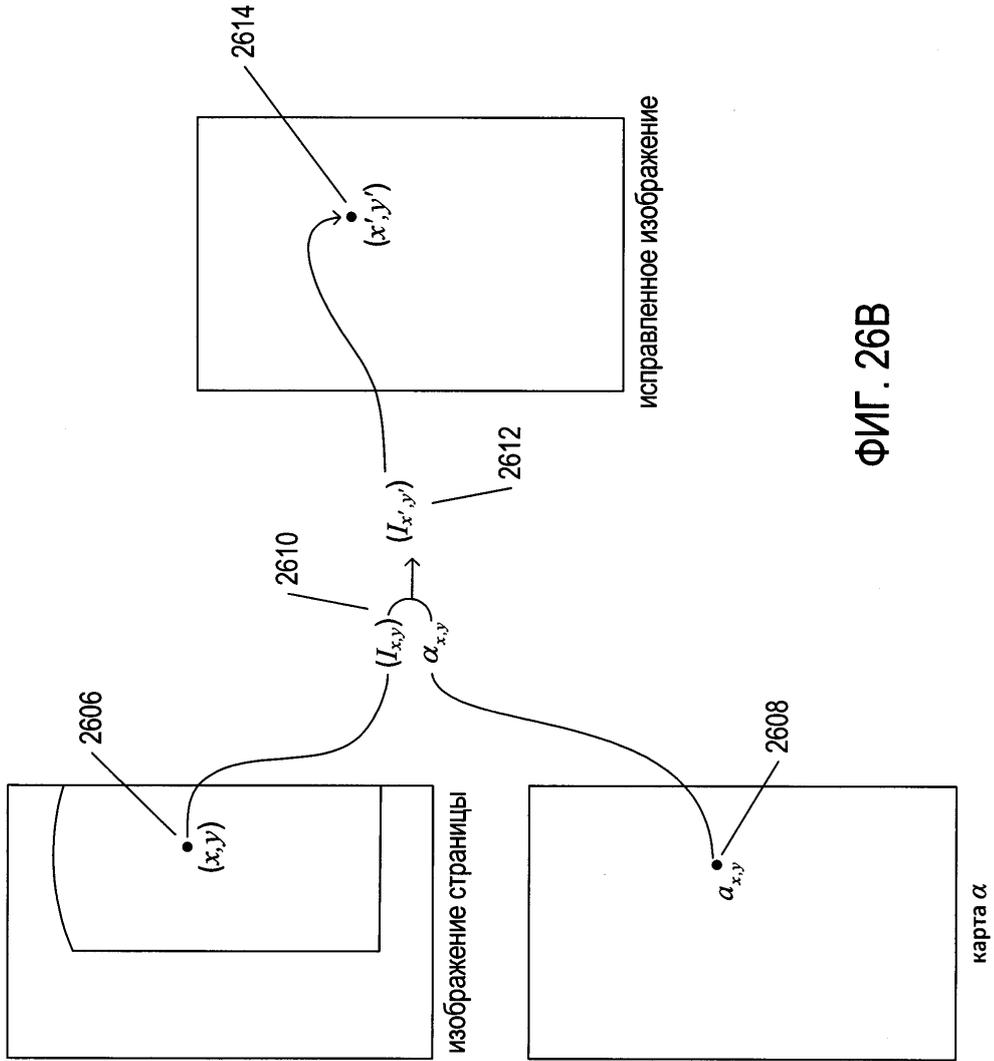
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

44/68



Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

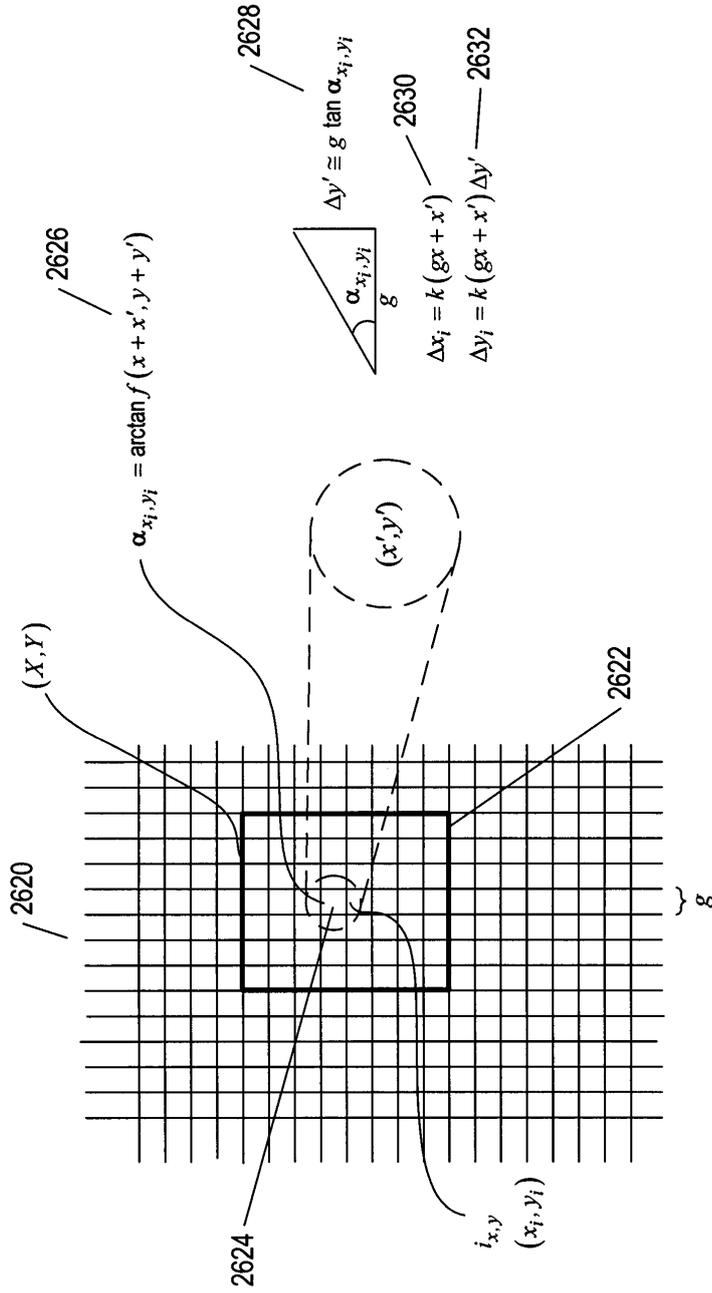
45/68



ФИГ. 26В

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

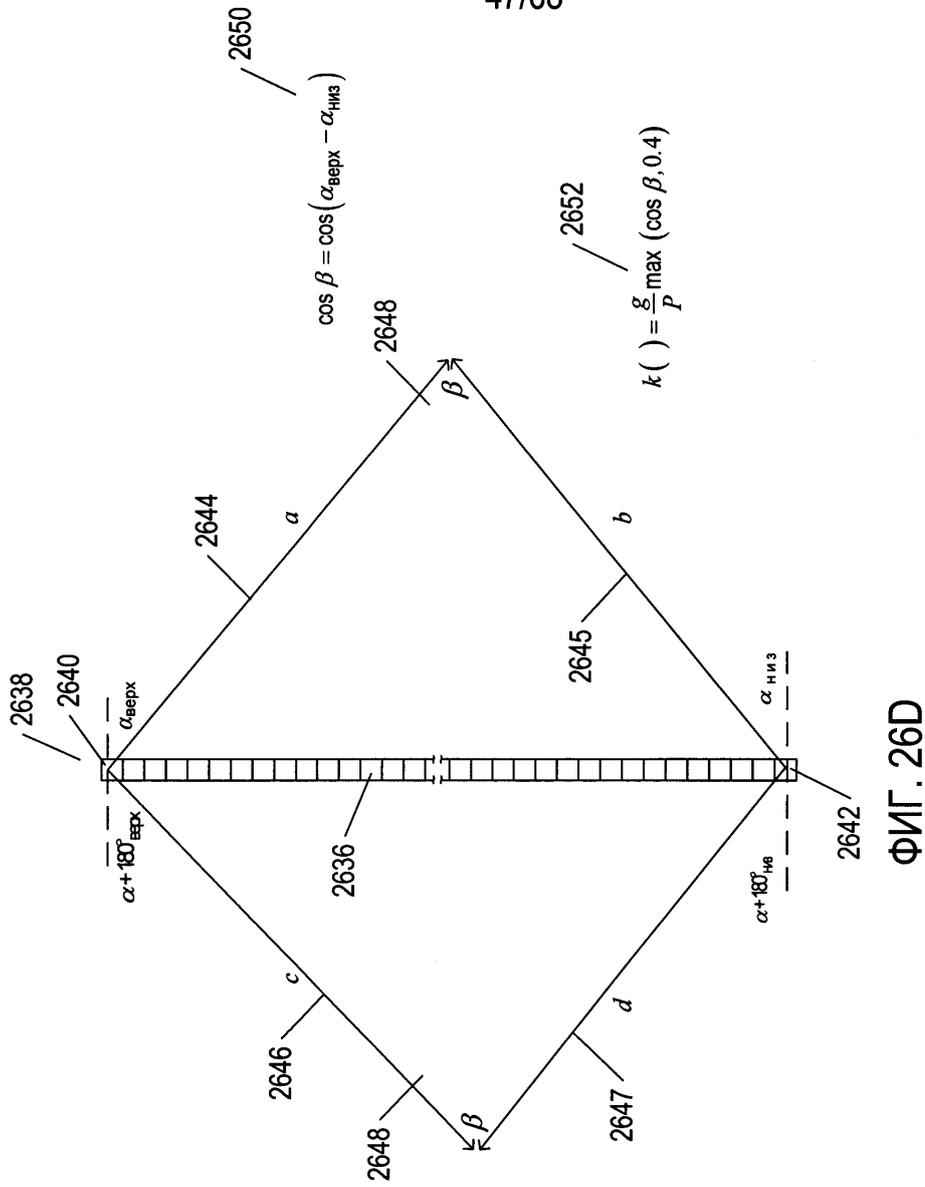
46/68



ФИГ. 26С

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

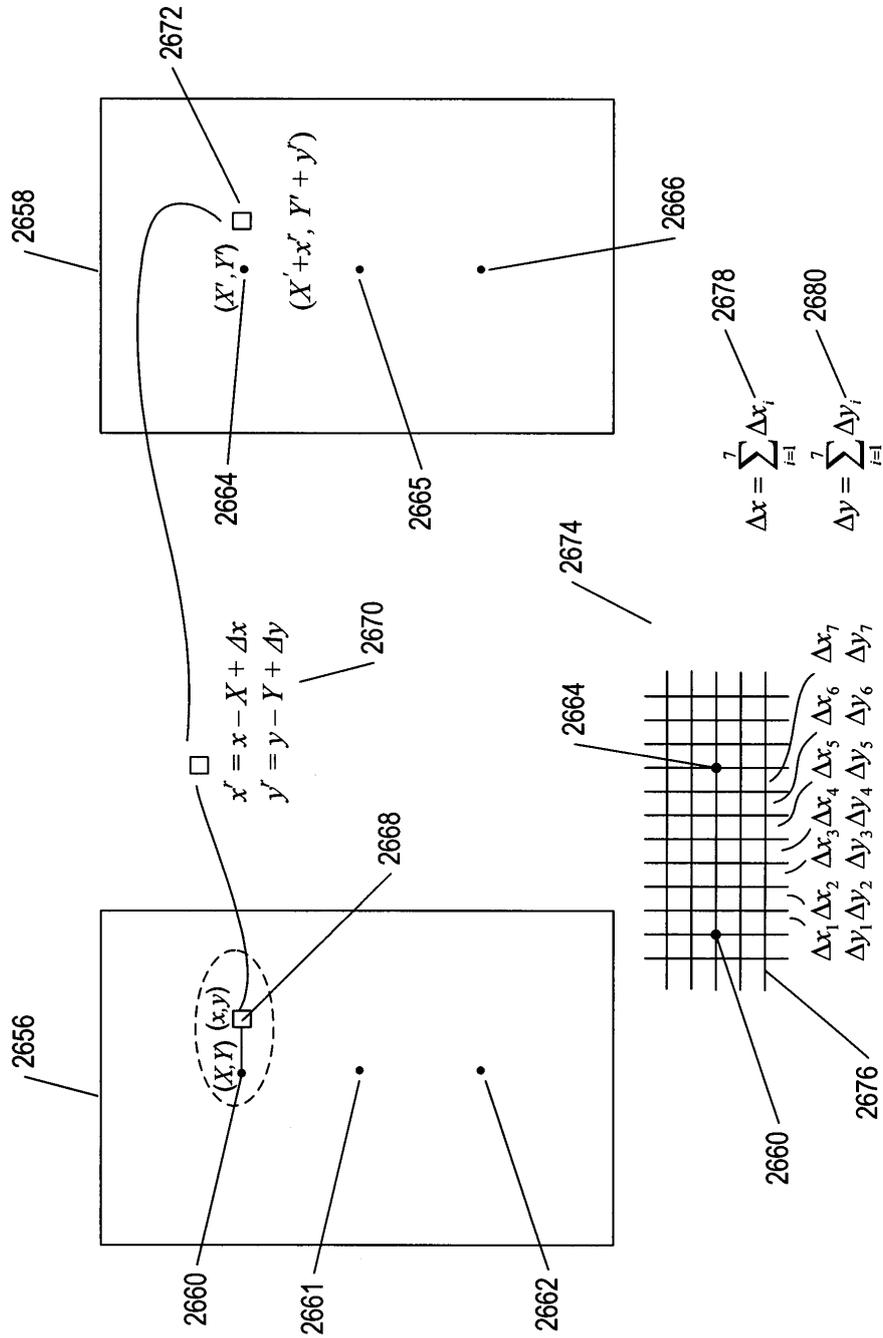
47/68



ФИГ. 26D

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

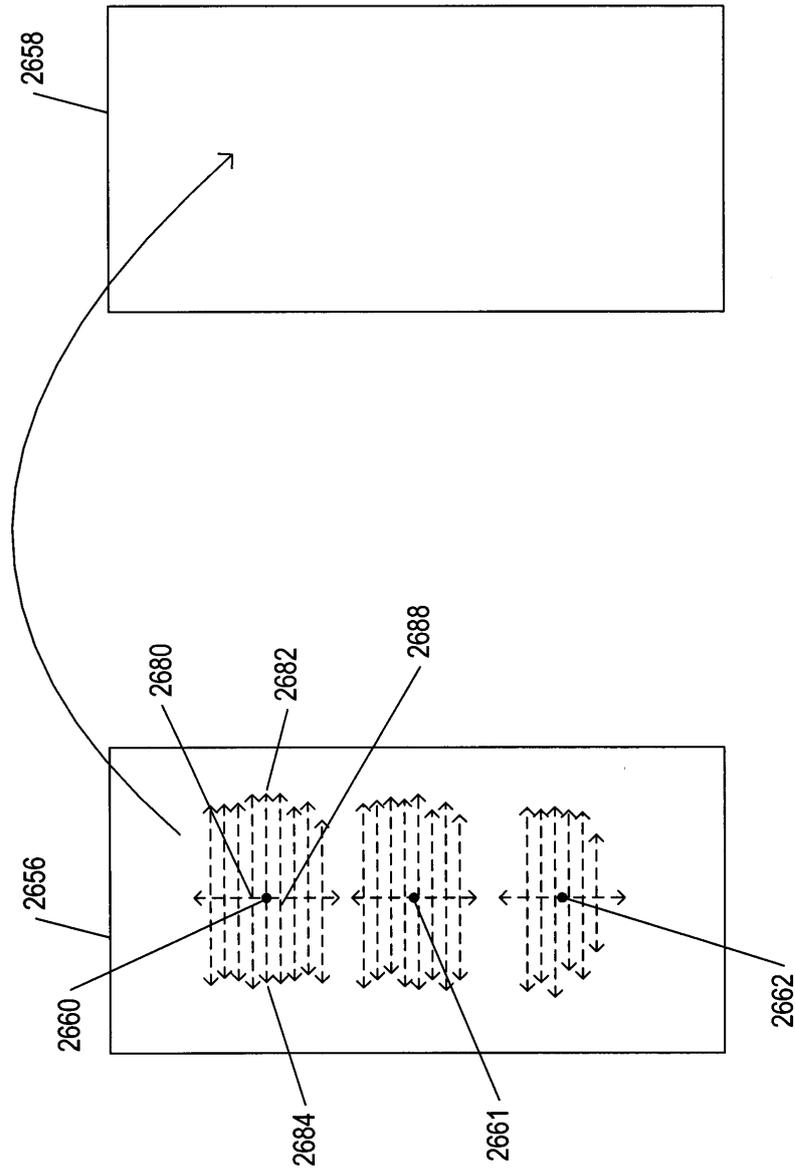
48/68



ФИГ. 26Е

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

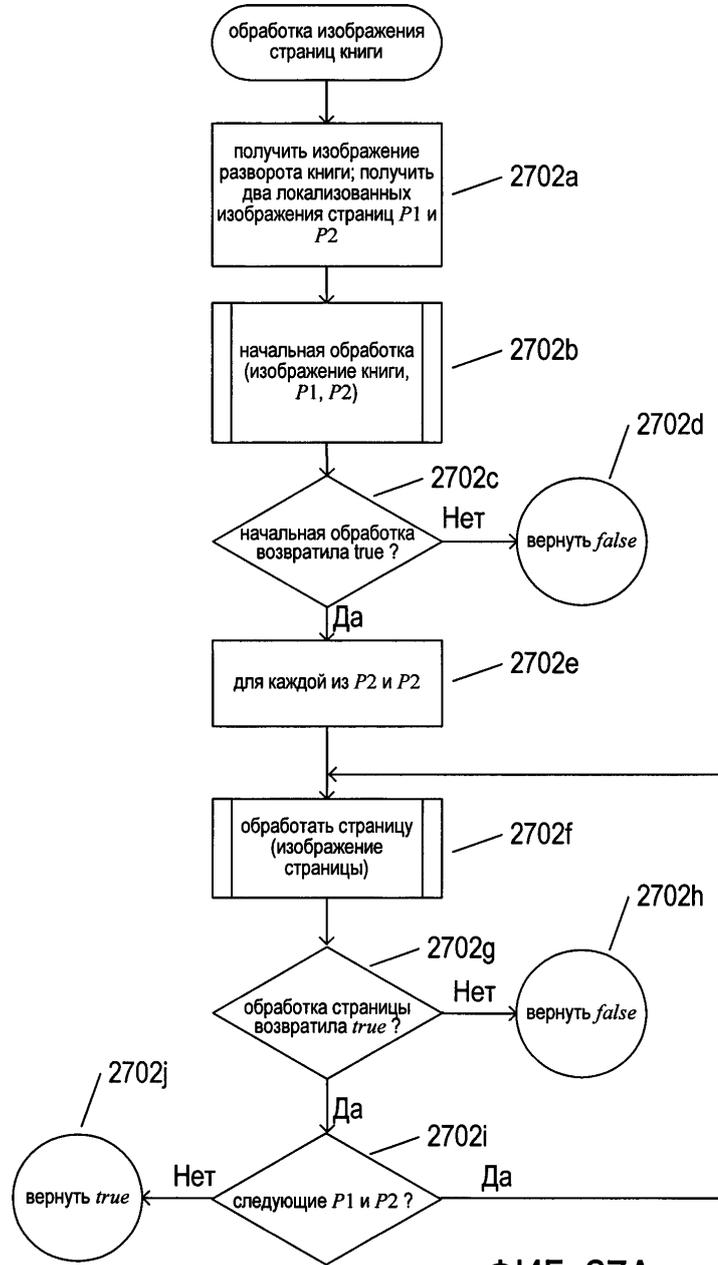
49/68



ФИГ. 26F

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АБВУУ 2014-IN-0198

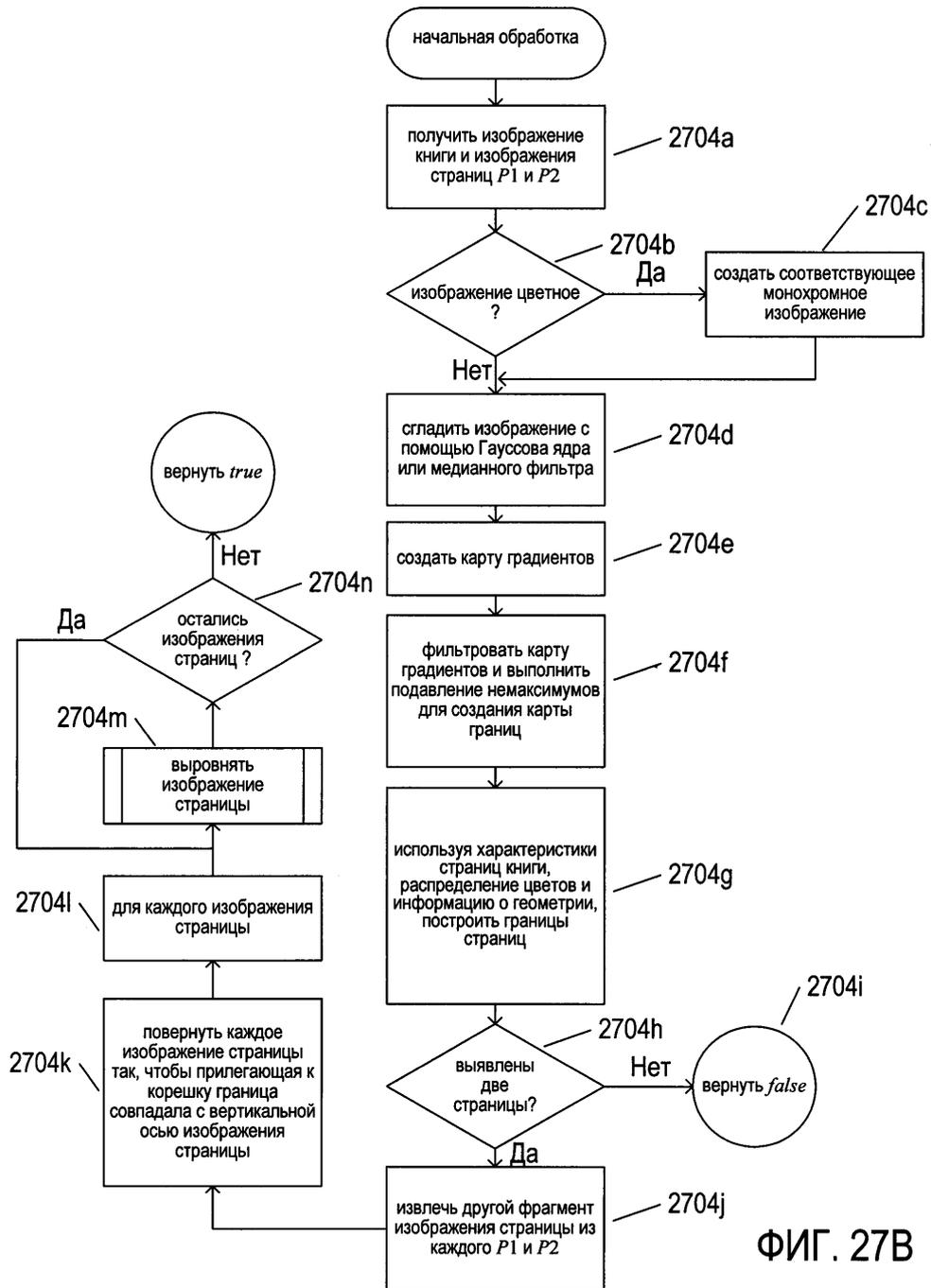
50/68



ФИГ. 27А

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АБВУУ 2014-IN-0198

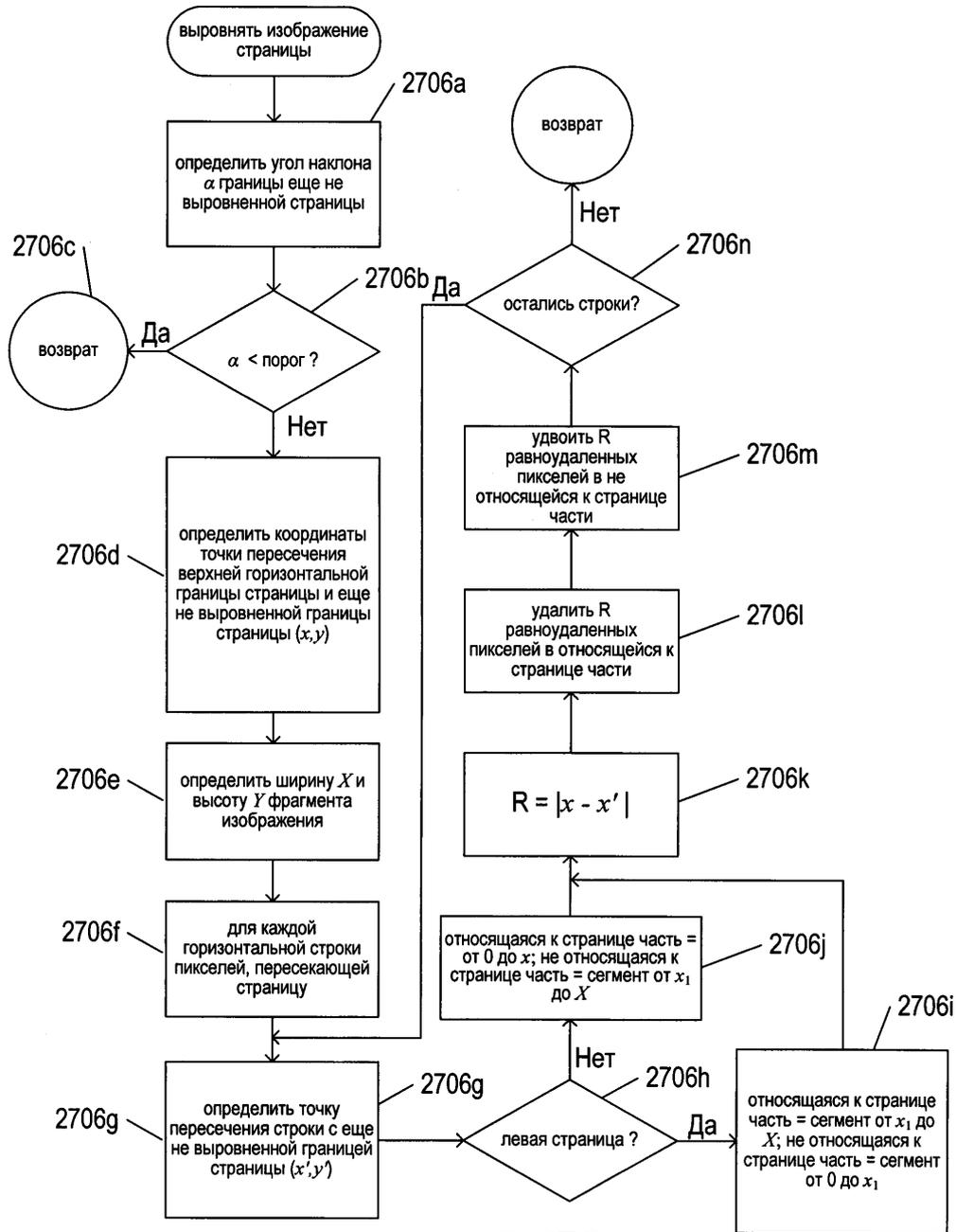
51/68



ФИГ. 27В

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

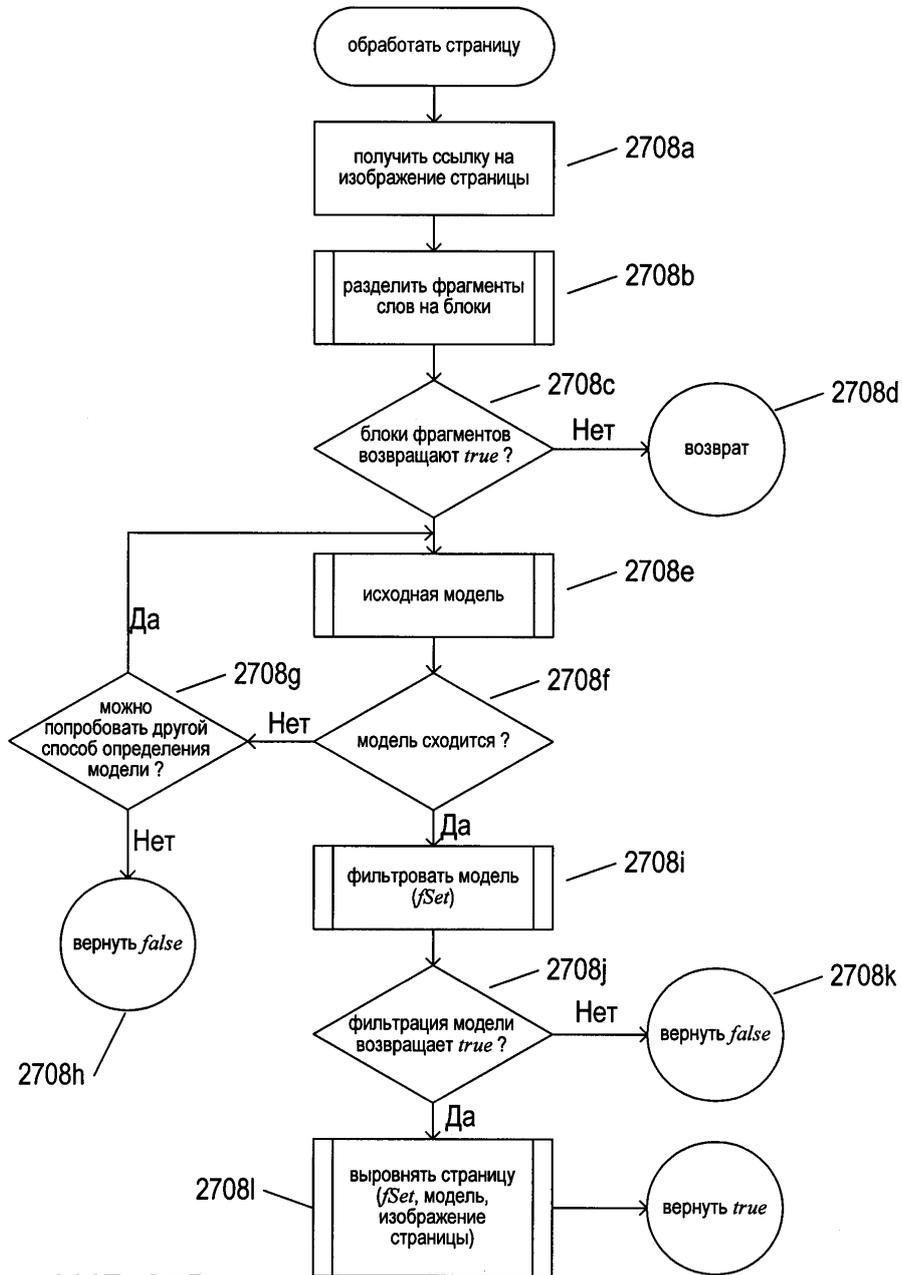
52/68



ФИГ. 27С

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

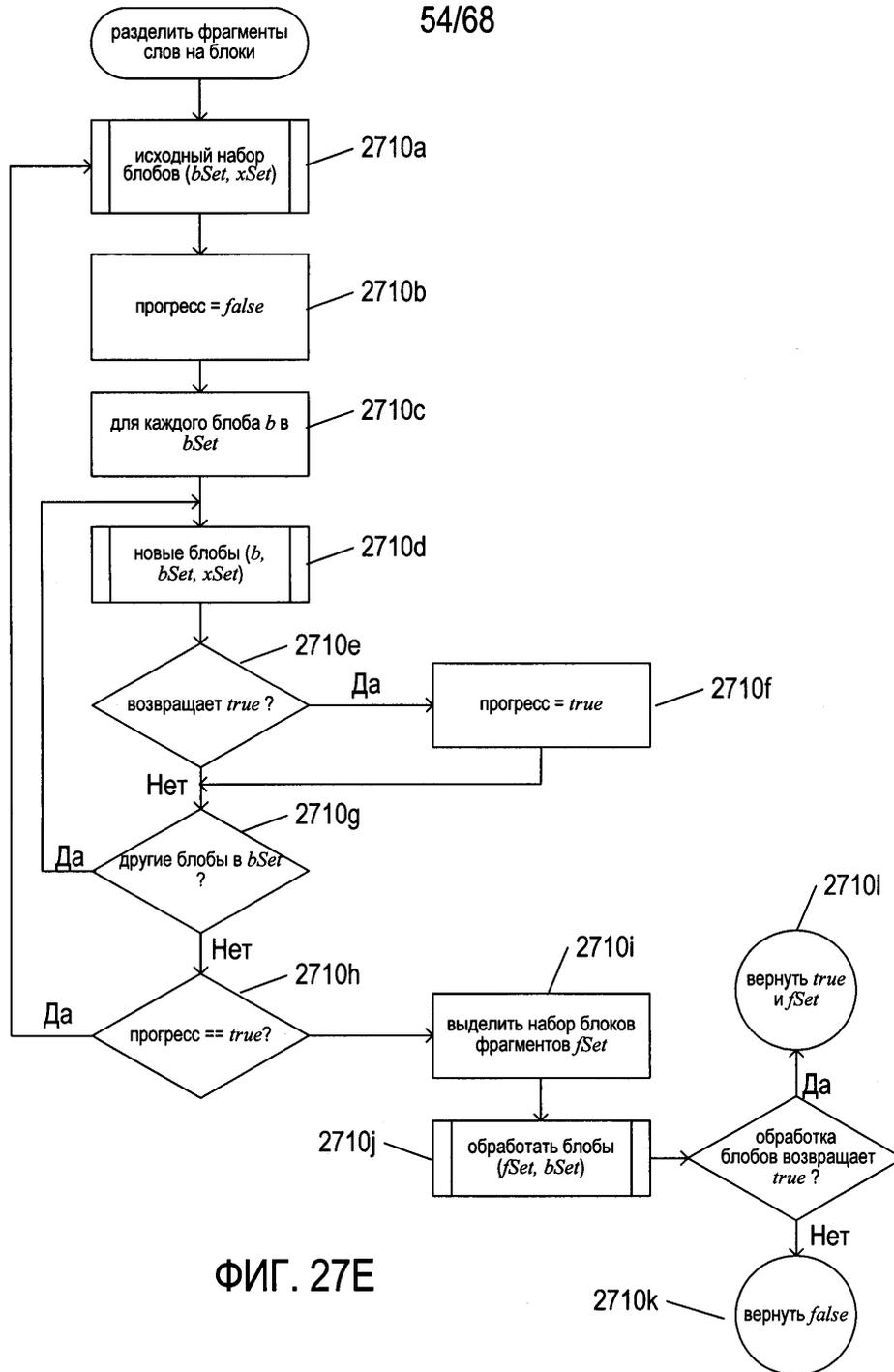
53/68



ФИГ. 27D

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АБВУУ 2014-IN-0198

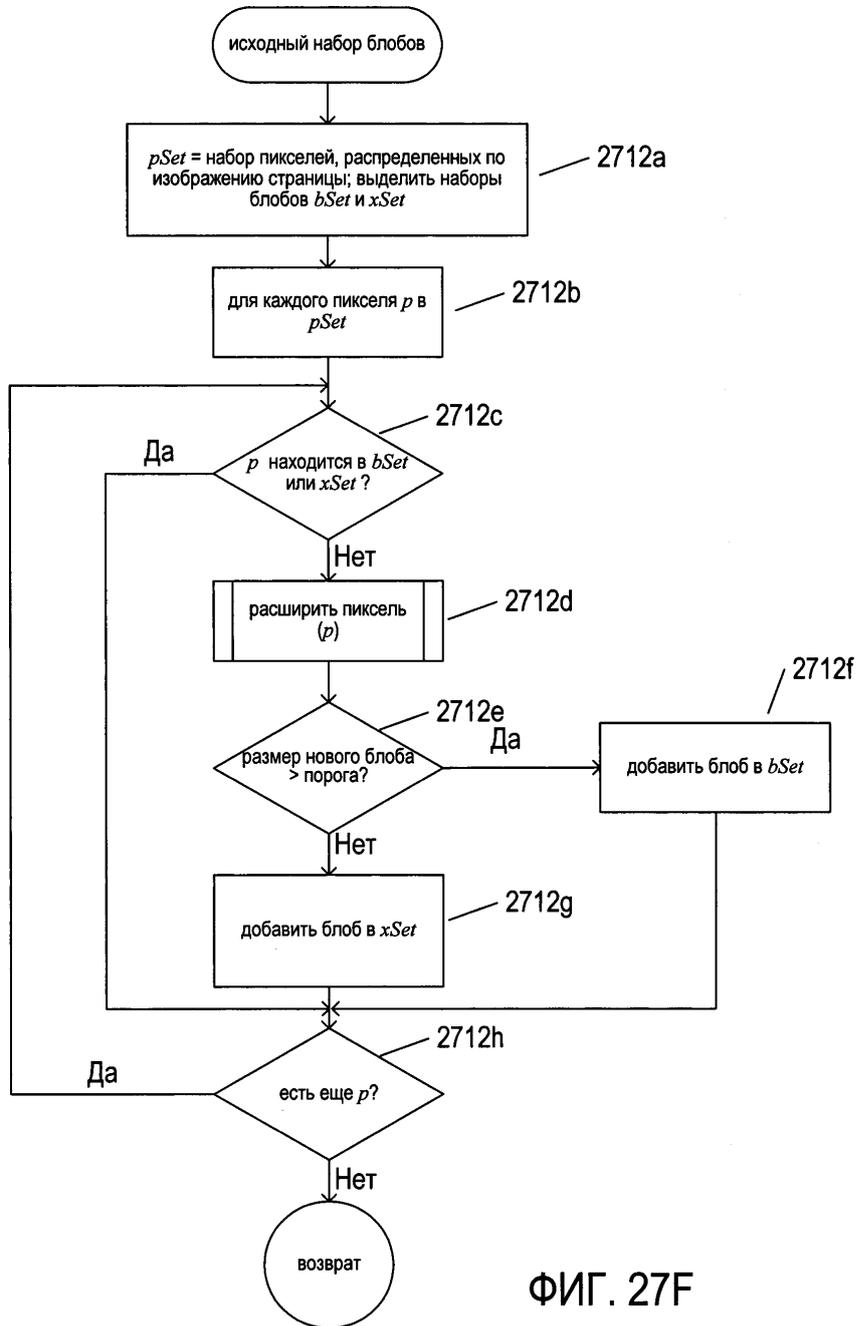
54/68



ФИГ. 27E

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложения/дело: АВВУУ 2014-IN-0198

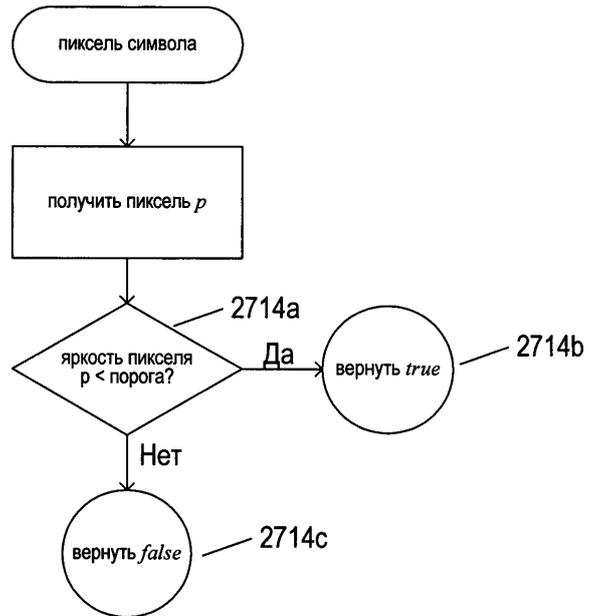
55/68



ФИГ. 27F

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

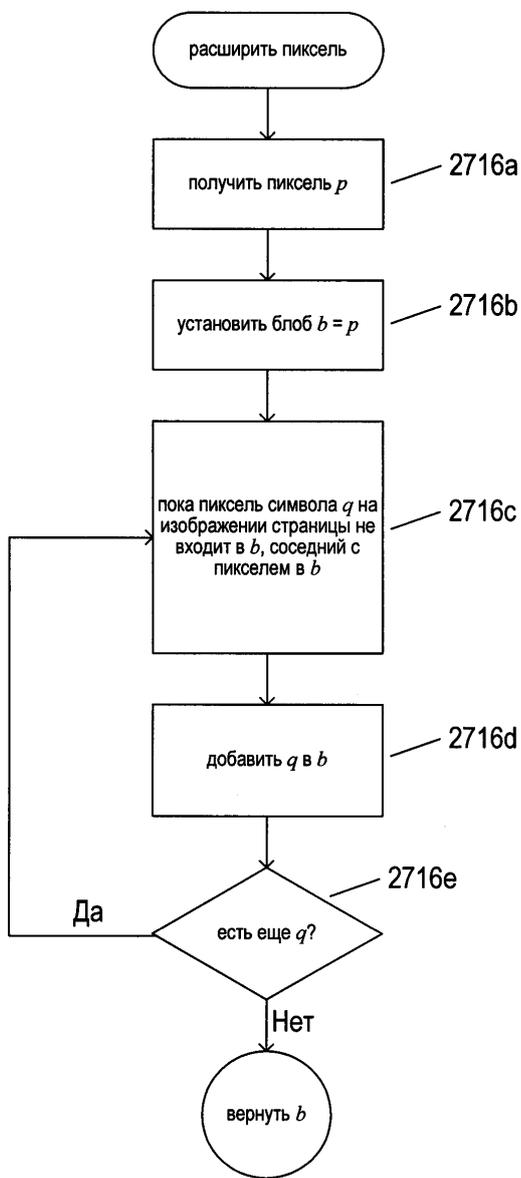
56/68



ФИГ. 27G

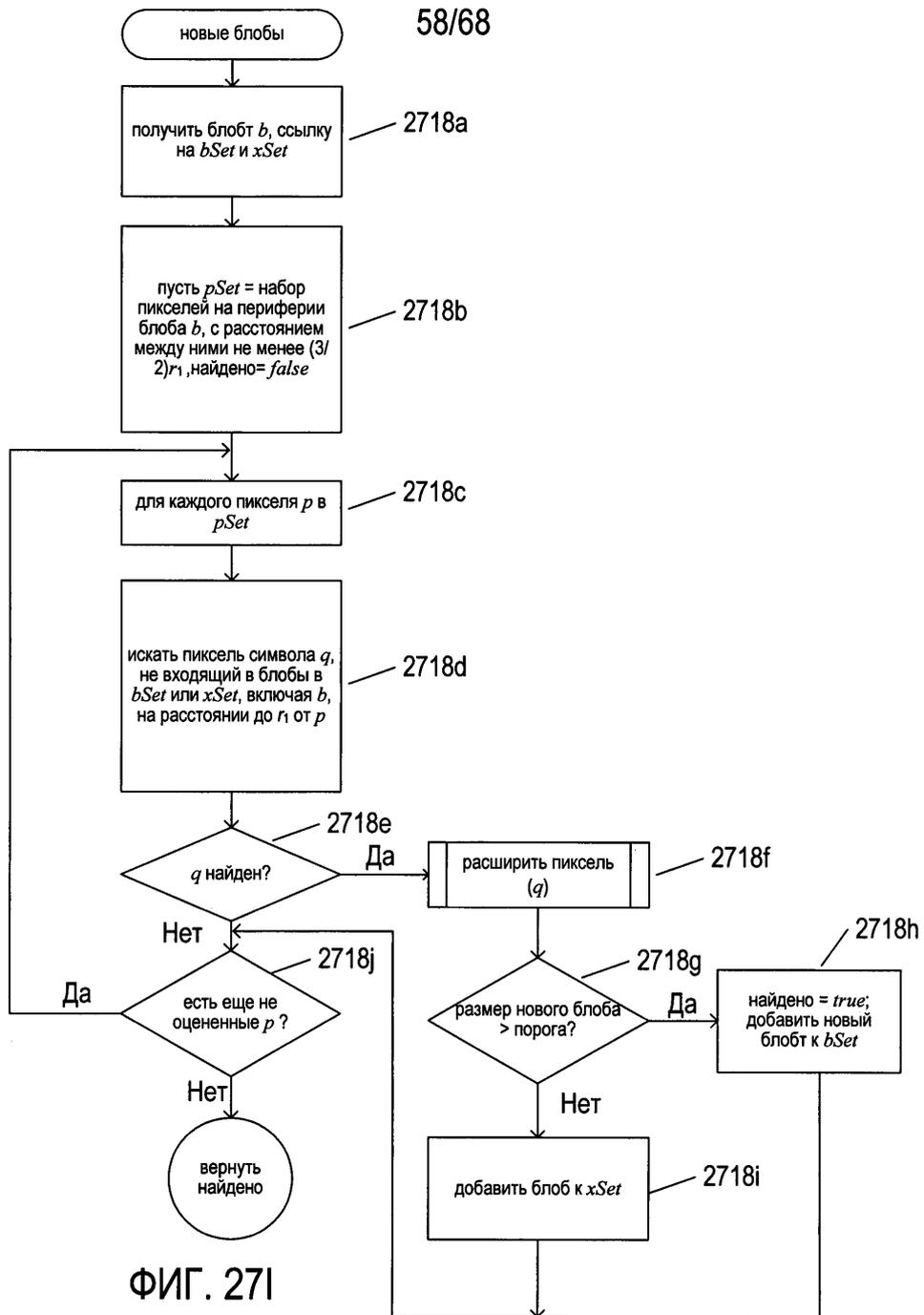
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

57/68

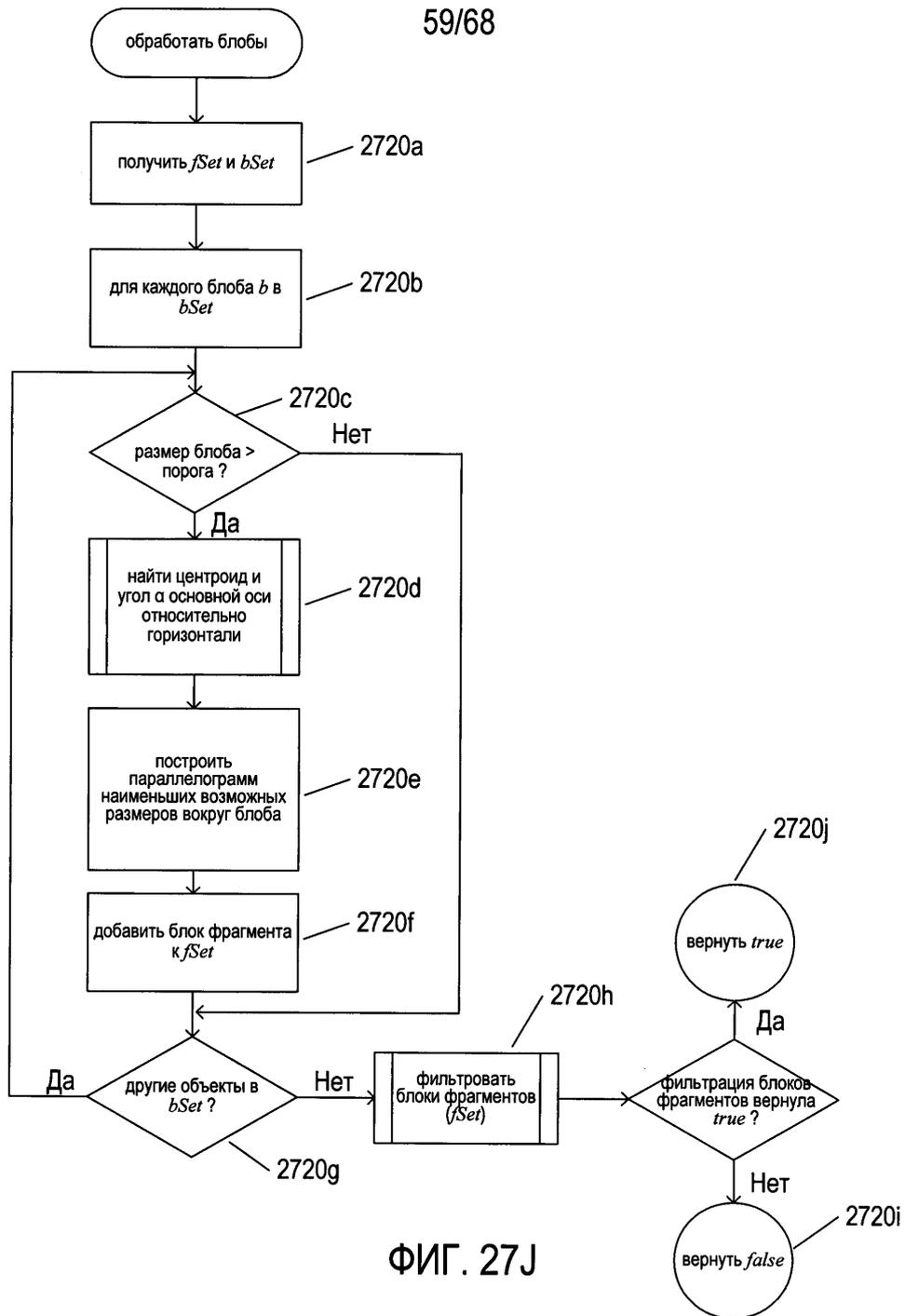


ФИГ. 27H

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

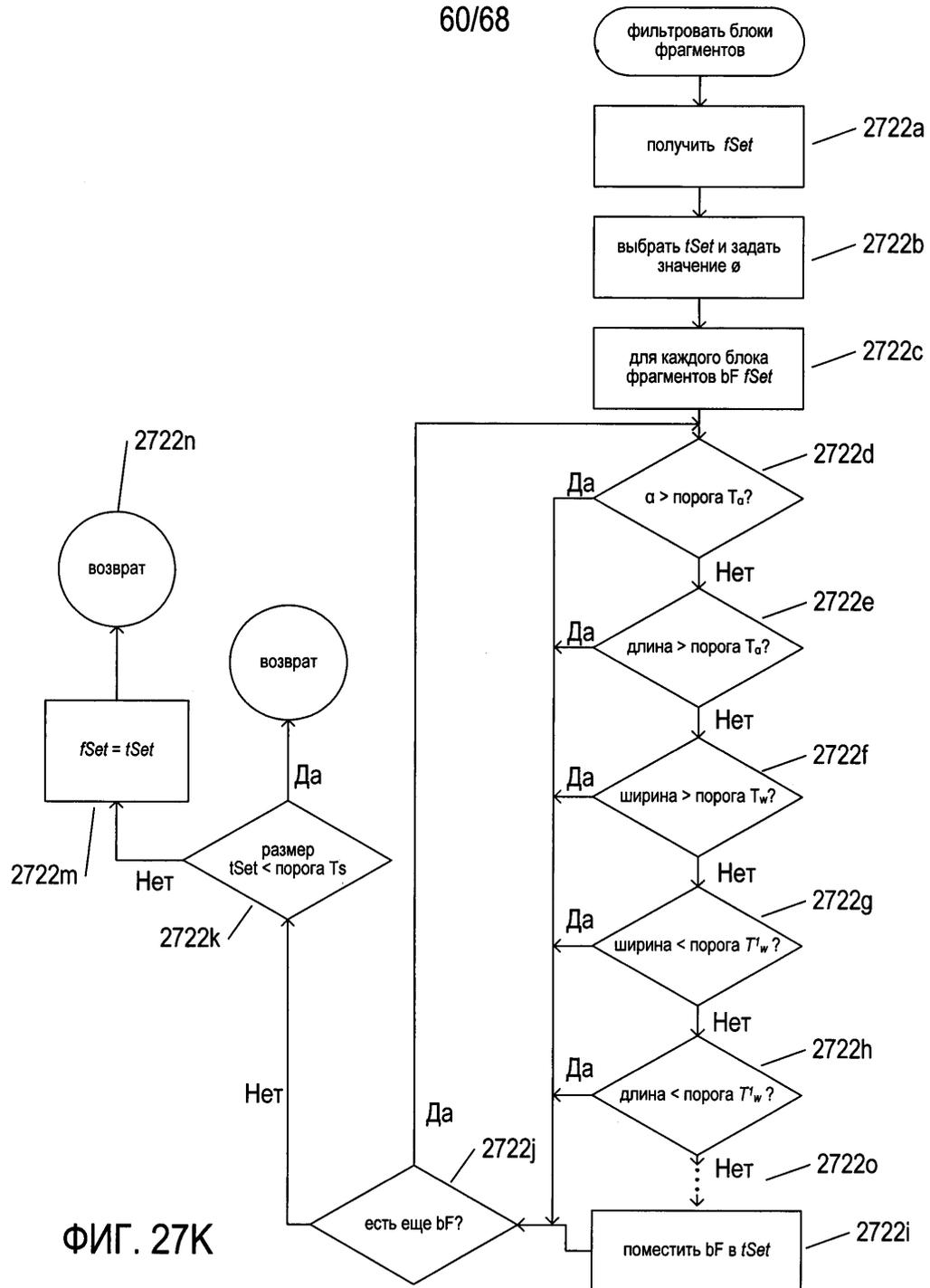


Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198



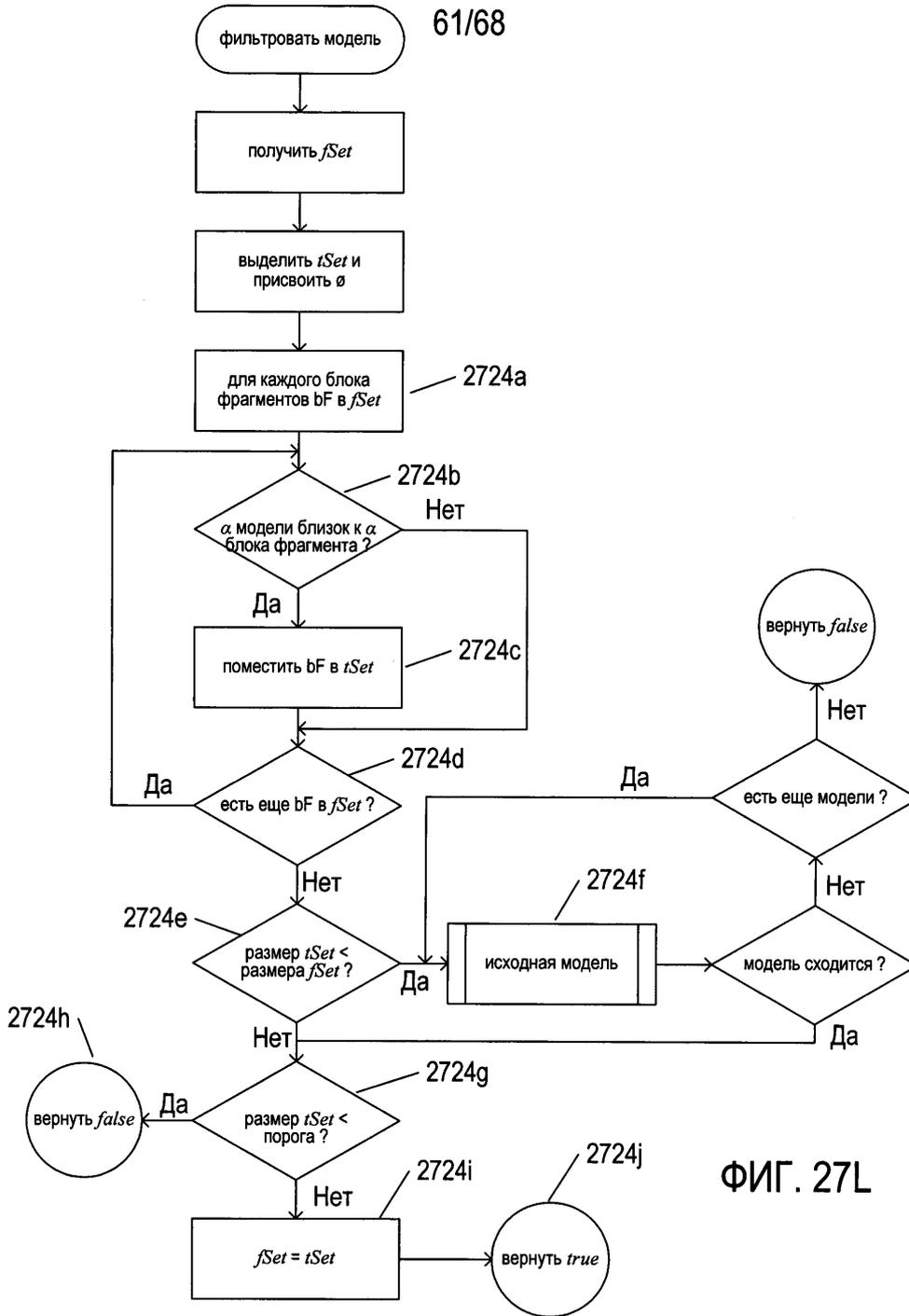
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

60/68



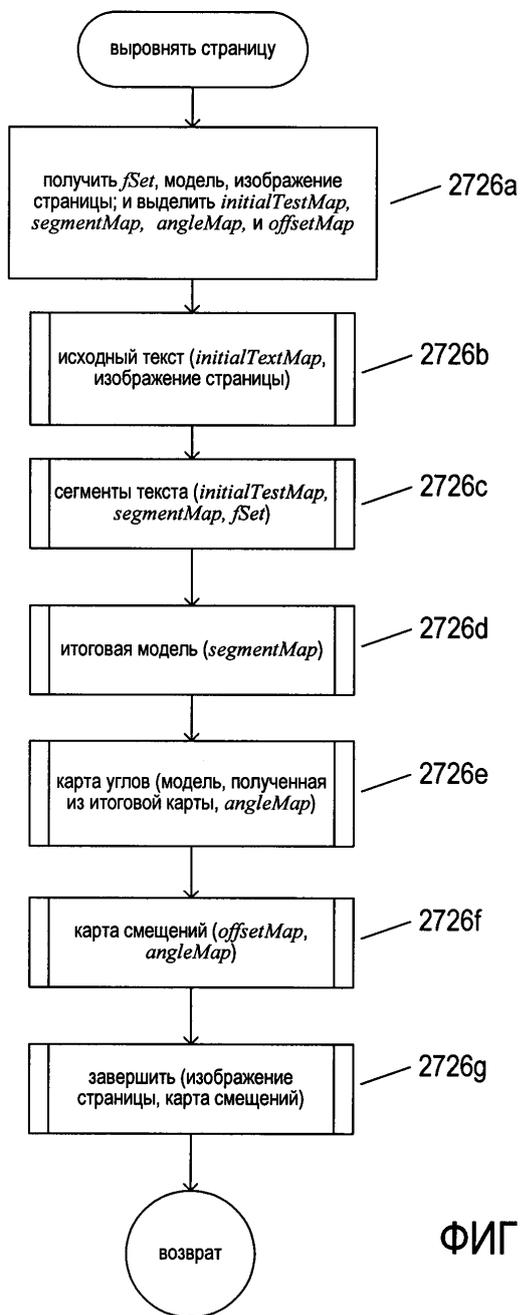
ФИГ. 27К

Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198



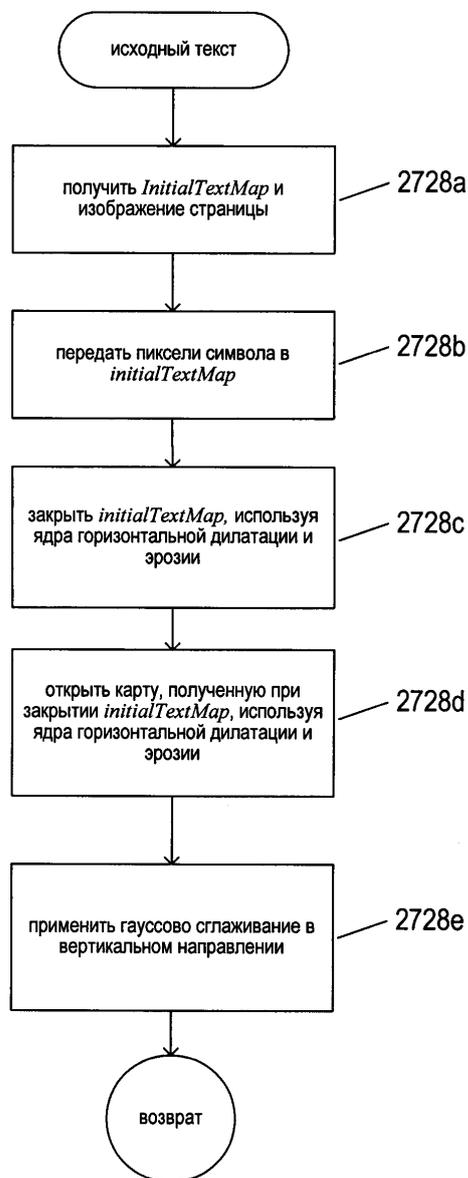
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АВВУУ 2014-IN-0198

62/68



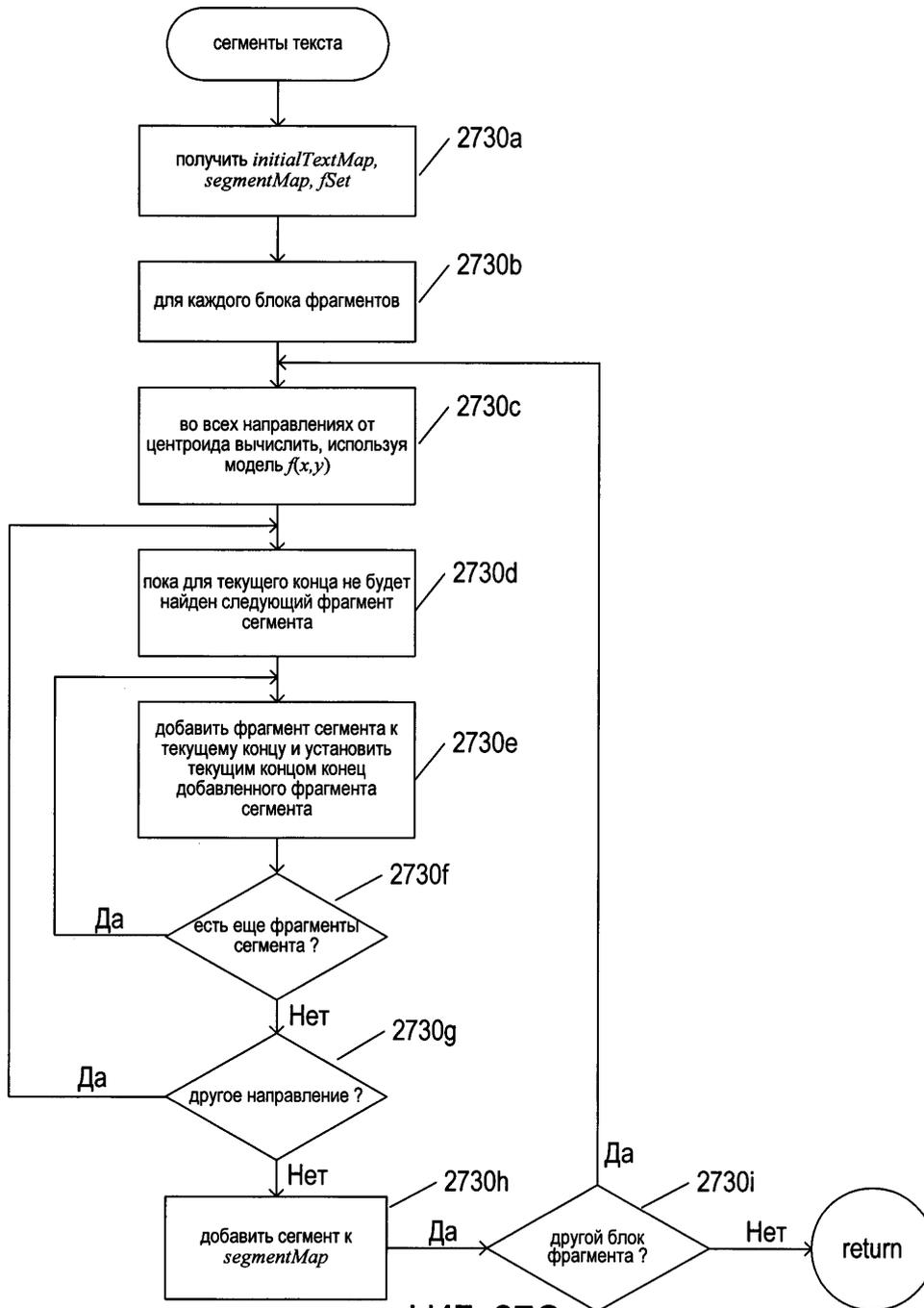
ФИГ. 27М

63/68



ФИГ. 27N

64/68



ФИГ. 270

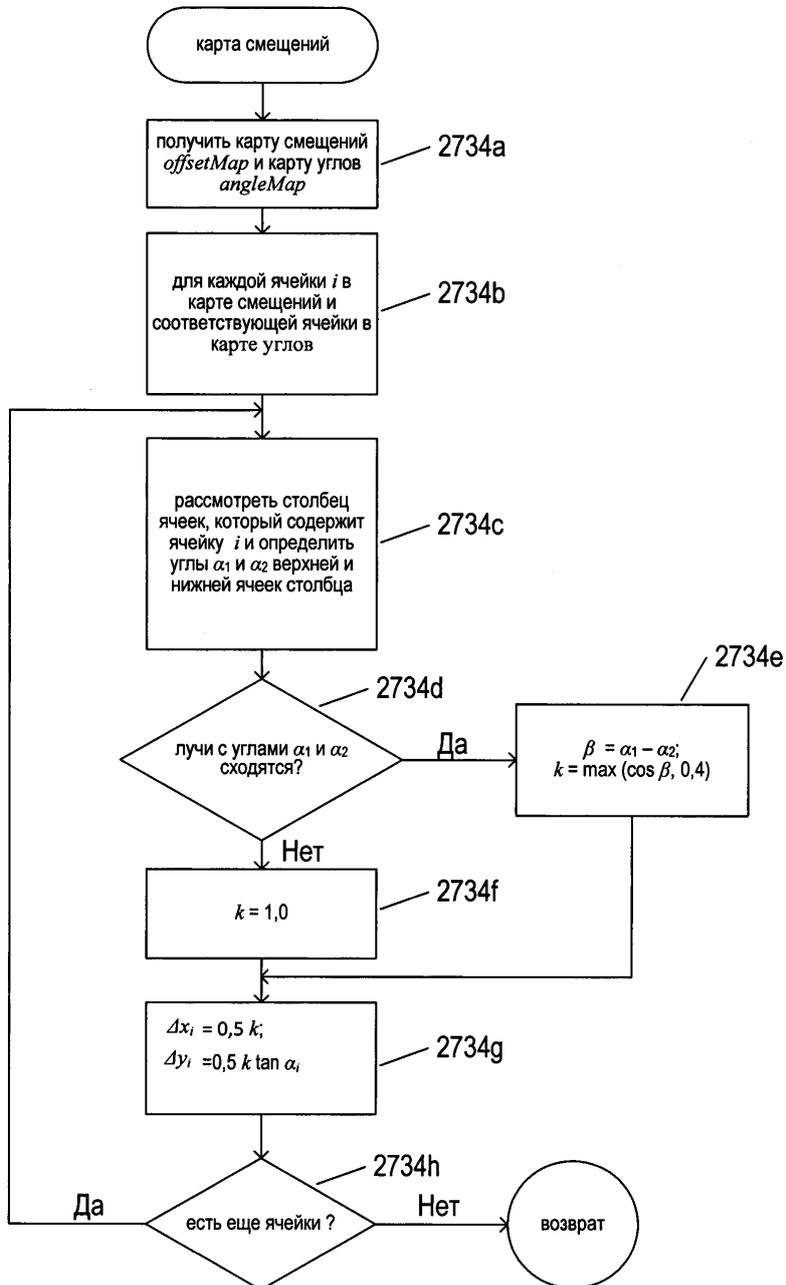
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АБВУУ 2014-IN-0198

65/68



ФИГ. 27Р

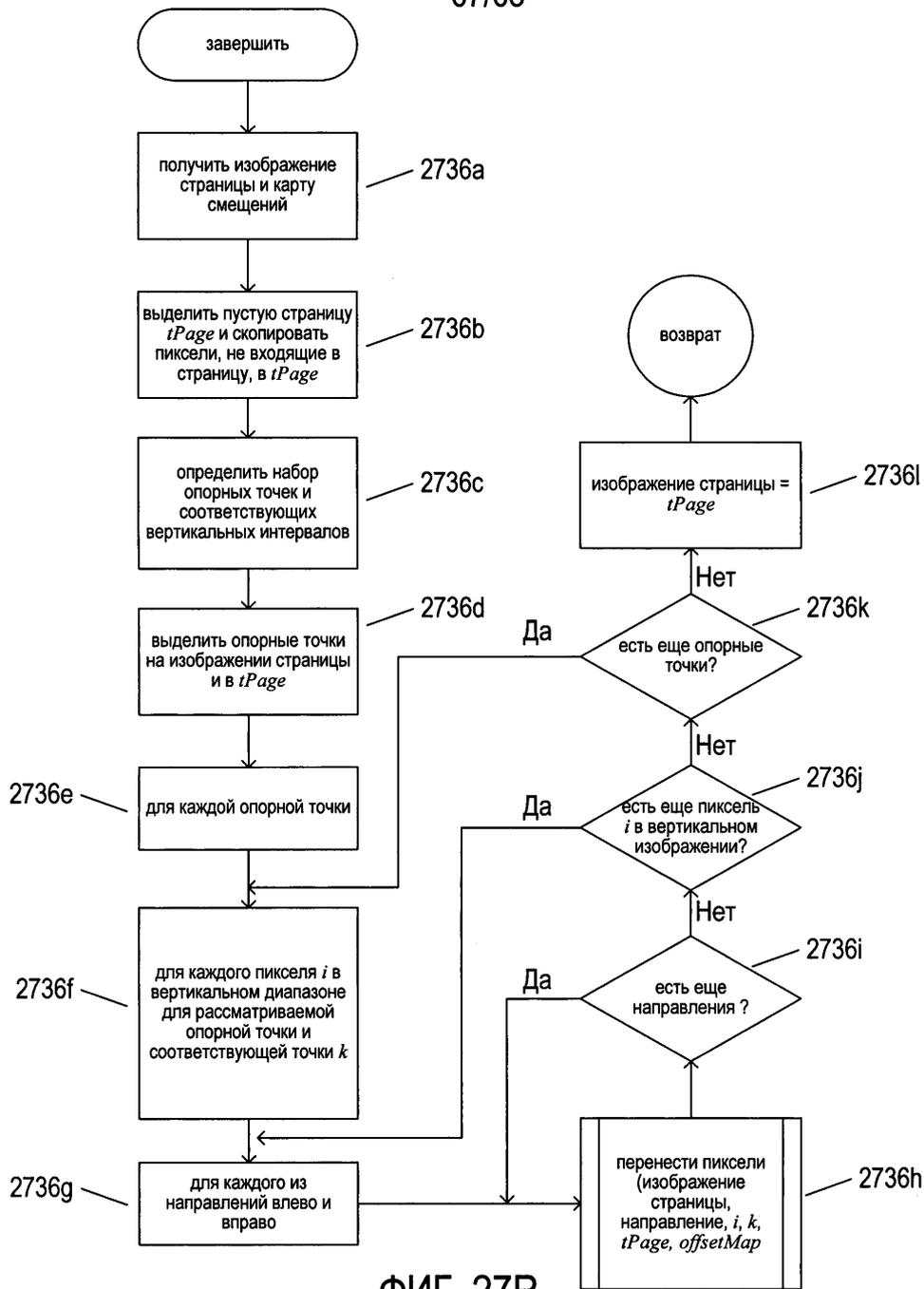
66/68



ФИГ. 27Q

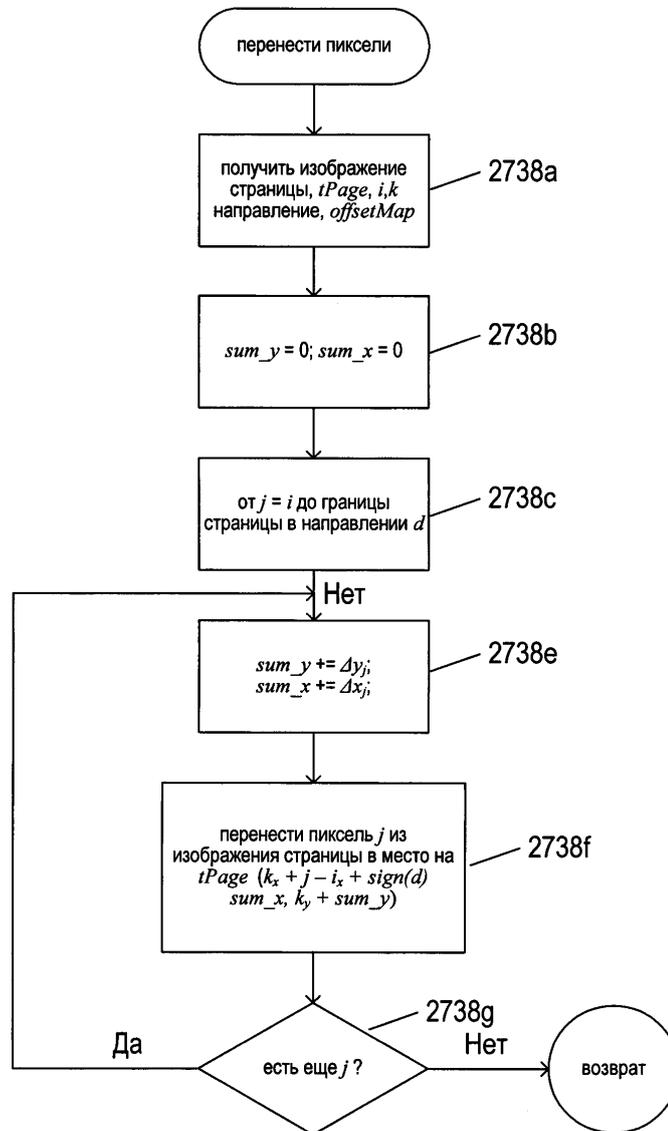
Название: СПОСОБ И СИСТЕМА ПОДГОТОВКИ  
СОДЕРЖАЩИХ ТЕКСТ ИЗОБРАЖЕНИЙ К  
ОПТИЧЕСКОМУ РАСПОЗНАВАНИЮ СИМВОЛОВ  
Автор изобретения:  
Приложение/дело: АБВУУ 2014-ИИ-0198

67/68



ФИГ. 27R

68/68



ФИГ. 27S