

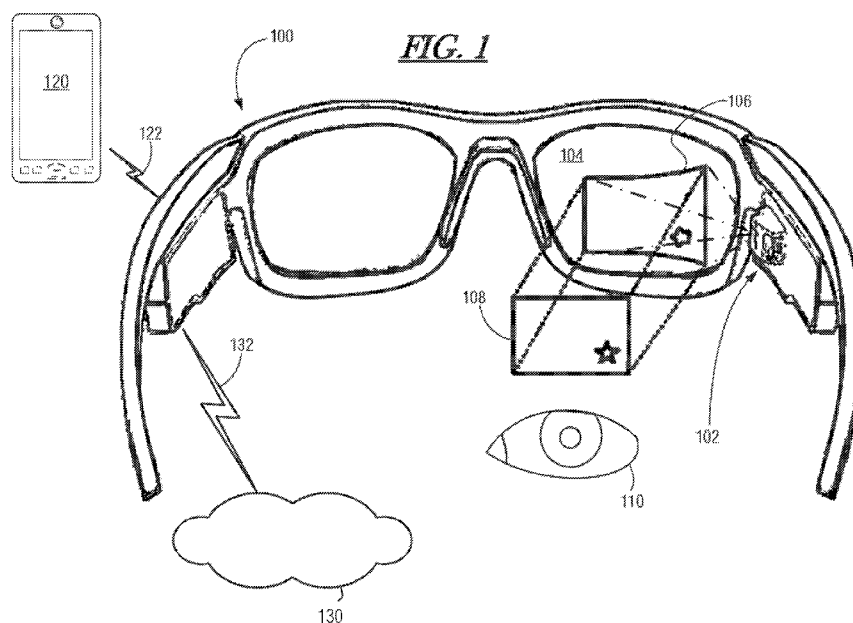


- (51) **International Patent Classification:**
G06T 3/00 (2006.01) *G06T 3/40* (2006.01)
G06T 5/00 (2006.01)
- (21) **International Application Number:**
PCT/US2016/061376
- (22) **International Filing Date:**
10 November 2016 (10.11.2016)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
14/757,868 23 December 2015 (23.12.2015) US
- (63) **Related by continuation (CON) or continuation-in-part (CIP) to earlier application:**
US 14/757,868 (CON)
Filed on 23 December 2015 (23.12.2015)
- (71) **Applicant: INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).
- (72) **Inventor: YEUNG, Fai;** 773 Newell Road, Palo Alto, California 94303 (US).
- (74) **Agents: GOULD, James R.** et al.; Schwegman Lundberg, & Woessner, P.A., P.O. Box 2938, Minneapolis, Minnesota 55402 (US).

- (81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— with international search report (Art. 21(3))

(54) **Title:** IMAGE PROCESSOR FOR WEARABLE DEVICE



(57) **Abstract:** Solutions for producing an image on an irregular surface are described. A graphical object is identified from an image to be displayed on the irregular surface. Objects according to at least one shape function are distorted to compensate for irregularities in the irregular surface. Previously-distorted instances of objects may be added to a distortion-compensated image.

IMAGE PROCESSOR FOR WEARABLE DEVICE

5 PRIORITY APPLICATION

[0001] This application claims the benefit of priority to U.S. Application Serial No. 14/757,868, filed on December 23, 2015, which is incorporated herein by reference in its entirety.

10 TECHNICAL FIELD

[0002] Embodiments described herein generally relate to an image processor and, more particularly, to an image processor to process graphical information for display on a surface having variable curvature, such as a smart glasses display.

15 BACKGROUND

[0003] The graphics rendering system in smart glasses may attempt to compensate for the curvature of lens before displaying an image on the lens. Without this compensation an image may appear to be distorted on smart glasses.

20 [0004] In a typical design, the processor (which may be local to the smart glasses, or remotely located, such as in the cloud, or in a companion smartphone, performs the computationally-intensive distortion compensation. In cases where the compensation processor is remote, its output, the distortion-compensated

25 image, is sent to the smart glasses. This remote-based processing has been relocated to the remote compensation processor in the companion phone, or elsewhere, because the local computing capability on the smart glasses tends to be limited by comparison. Still, the compensation for distortion is not a trivial computational problem even for the companion phone.

30 [0005] Previous solutions for distortion compensation involve direct pixel interpolation of the entire image via linear transforms such as perspective transform. While perspective transform may reasonably describe linear space, it may not fit the curvature of the lens very well because the lens requires higher order of description for its curved surface.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] In the drawings, which are not necessarily drawn to scale, like numerals may describe similar components in different views. Like numerals having different letter suffixes may represent different instances of similar components.

5 Some embodiments are illustrated by way of example, and not limitation, in the figures of the accompanying drawings.

[0007] FIG. 1 is a diagram illustrating the general operation of displaying an image onto the lens of a pair of smart glasses, as well as optional distributed processing devices that may work with the smart glasses according to some
10 embodiments.

[0008] FIG. 2 is a block diagram illustrating an exemplary system architecture of a computing device, such as one or more of the computing devices depicted in FIG. 1, according to an embodiment.

[0009] FIG. 3 is a diagram illustrating an exemplary hardware and software
15 architecture of a computing device such as the one depicted in FIG. 2, in which various interfaces between hardware components and software components are shown.

[0010] FIG. 4 is a system block diagram illustrating a system for rendering objects to be displayed on an irregular surface according to some embodiments.

20 [0011] FIG. 5 is a diagram illustrating a transformation of a Lagrange parent element to an irregular Lagrange element according to some embodiments.

[0012] FIGs. 6 and 7 are a diagrams illustrating a Lagrange parent element, which may be discretized according to some embodiments.

[0013] FIG. 8 illustrates an example of a transformation of a textual object
25 according to some embodiments.

[0014] FIG. 9 is a diagram illustrating a process of modeling the visual objects to be displayed using a deformable mesh according to some embodiments.

[0015] FIG. 10 is a flow diagram illustrating an example of a process to correct
30 an image for display on a surface having an irregular geometry according to some embodiments.

DETAILED DESCRIPTION

[0016] Aspects of the embodiments are directed to distortion compensation for images to be displayed on a curved surface. While conventional approaches,

such as those summarized above, for instance, operate on all pixels of the image each time the display is refreshed, some of the present embodiments recognize that this conventional brute-force approach is unnecessary because, instead of the entire image, only some visual elements may be subject to updating. For example, while an on-screen clock updates the time, the rest of screen elements such as location, upcoming calendar appointment, etc. remain unchanged.

[0017] In some embodiments, screen rendering is computed based on a deformable mesh. Accordingly, the visual content is modeled as deformable objects with specific deformation characteristics. In some examples, the deformation associated with each object may be described by the shape function of a Lagrange element representing the object. Each Lagrange element represents a second-order non-rigid surface. As a result, this type of modeling of deformation fits the distortion of the lens more precisely than linear models using affine or perspective transform, for example.

[0018] In related embodiments, deformable objects are updated independently. Advantageously, the update of an object avoids having to compute updates of the entire image, thus significantly reducing the overall computational load.

[0019] FIG. 1 is a diagram illustrating the general operation of displaying of an image onto the lens of a pair of smart glasses 100 according to some embodiments. As depicted, smart glasses 100 include a display projector 102 that projects an image onto an interior surface 104 of the lens. Interior surface 104 has a complex shape that includes varying degrees of curvature. Image 106 is projected by display projector 102 onto lens surface 104 in a distortion-compensated format, as illustrated, to be perceived by the user 110 as a non-distorted image 108.

[0020] It should be noted that smart glasses 100 may take a variety of different form factors, and may generally include various other components, including computing hardware, power circuitry, user input and various sensing/data capture devices, such as a camera, microphone, touch sensor, etc., as well as wireless networking interface devices, and the like, some of which are described in greater detail below.

[0021] Smart glasses 100 may include a wireless interface for communicating with a companion smartphone 120 via wireless link 122. For instance, a Bluetooth Low-Energy (BLE) connection may be supported. In some related

embodiments, smart glasses 100 and companion smartphone 120 may use distributed computing techniques to off-load certain computing operations to smartphone 120, which generally has more computing capacity and battery capacity than smart glasses 100. In other embodiments, smart glasses 100 may
5 connect wirelessly to remote service 130, which may be a cloud-based computing model, via wireless link 132. Wireless link 132 may be wireless to a base station, such as a Wi-Fi base station, or a Long-Term Evolution (LTE) e-node B (neither of which is shown for the sake of clarity), which in turn connects to service provider 130 via a backhaul or Internet service provider's wide-area
10 network (WAN).

[0022] FIG. 2 is a block diagram illustrating an exemplary system architecture 200 of smart glasses 100 or smartphone 120, each of which may generally be considered a computing device. Depending on the form factor and nature of the smart glasses 100 or smartphone 120 computing devices, not every component
15 described below may be present. Central processor unit (CPU) 202 includes one or more microprocessors on which the overall functionality of computing device 100 is executed. CPU 202 is formed from hardware that is electrically interfaced with system link 203, which carries data and control signaling between the various components. As illustrated, system link 203 is similarly interfaced with
20 each of the other components of system architecture 200. Memory 204 includes working memory space, and is constructed from suitable high-speed memory devices such as synchronous dynamic random access memory (SDRAM). In the embodiment illustrated, CPU 202 may access memory 204 using high-speed interface 205. Non-volatile memory 206 is constructed using read-only memory
25 (ROM), electrically-erasable programmable read-only memory (EEPROM), flash memory or other suitable non-volatile storage technology. Non-volatile memory 206 stores system and application software that is executed by CPU 202 and, in some cases, by processors present in one or more other components.

[0023] External non-volatile memory 207, where available, includes an
30 interface such as a secure digital (SD) card slot, which may accept removable storage media to be used as additional non-volatile data storage.

[0024] Display 208 includes display 104 and circuitry for interfacing the display 104 with the system, as well as video driving circuitry and, in the case of smart glasses 100, a display projection device. Sound 210 contains circuitry for

driving the audio output to a speaker or headphones, and the circuitry for interfacing with the system. User input 212 contains the circuitry for interfacing with input devices such as a tapping gesture sensor or touchscreen input device. Communications block 214 represents communications circuitry and circuitry for interfacing the communications circuitry with the system. Communications block 214 may include a radio for communicating over a cellular network such as a network designed according to the LTE, LTE-Advanced, 5G or Global System for Mobile Communications (GSM) families of standards. Also, communications circuitry 214 may include a Wi-Fi communications radio according to the IEEE 801.11 family of standards, or a Bluetooth radio circuit according to the IEEE 802.15 family of standards. Real-time clock 216 includes circuitry that provides a clock that maintains the current date and time, and that interfaces the clock to the system.

[0025] Data capture devices 220 are integrated with computing device 200. According to various embodiments, data capture devices 220 include a plurality of different types of sensing transducers and their associated processing and interface circuitry, such as an accelerometer, as well as a camera, GPS, and biometric sensors according to various embodiments.

[0026] The processing circuitry associated with each corresponding transducer may include amplification, buffering, filtering, or other signal-conditioning circuitry to receive the raw analog signal from the corresponding transducer and prepare the analog signaling for digitization, analog-to-digital conversion circuitry to perform sampling, quantization, and digital encoding, and, in some cases, further processing to produce a digital signal representing the physical phenomenon being measured by the transducer in a form that is readable by CPU 202.

[0027] FIG. 3 is a diagram illustrating an exemplary hardware and software architecture of a computing device such as the one depicted in FIG. 2, in which various interfaces between hardware components and software components are shown. As indicated by HW, hardware components are represented below the divider line, whereas software components denoted by SW reside above the divider line. On the hardware side, processing devices 302 (which may include one or more microprocessors, digital signal processors, etc., each having one or more processor cores, are interfaced with memory management device 304 and

system interconnect 306. Memory management device 304 provides mappings between virtual memory used by processes being executed, and the physical memory. Memory management device 304 may be an integral part of a central processing unit which also includes the processing devices 302.

- 5 **[0028]** Interconnect 306 includes a backplane such as memory, data, and control lines, as well as the interface with input/output devices, e.g., proprietary bus, PCI, USB, etc. Memory 308 (e.g., dynamic random access memory - DRAM) and non-volatile memory 309 such as flash memory (e.g., electrically-erasable read-only memory – EEPROM, NAND Flash, NOR Flash, etc.) are
10 interfaced with memory management device 304 and interconnect 306 via memory controller 310. This architecture may support direct memory access (DMA) by peripherals in some embodiments. I/O devices, including video and audio adapters, non-volatile storage, external peripheral links such as USB, Bluetooth, etc., as well as network interface devices such as those
15 communicating via Wi-Fi or LTE-family interfaces, are collectively represented as I/O devices and networking 312, which interface with interconnect 306 via corresponding I/O controllers 314.

- [0029]** On the software side, a pre-operating system (pre-OS) environment 316, which is executed at initial system start-up and is responsible for initiating
20 the boot-up of the operating system. One traditional example of pre-OS environment 316 is a system basic input/output system (BIOS). In present-day systems, a unified extensible firmware interface (UEFI) may be implemented. Operating system (OS) 318 provides a kernel that controls the hardware devices, manages memory access for programs in memory, coordinates tasks and
25 facilitates multi-tasking, organizes data to be stored, assigns memory space and other resources, loads program binary code into memory, initiates execution of the application program which then interacts with the user and with hardware devices, and detects and responds to various defined interrupts. Also, operating system 318 provides device drivers, and a variety of common services such as
30 those that facilitate interfacing with peripherals and networking, that provide abstraction for application programs so that the applications do not need to be responsible for handling the details of such common operations. Operating system 318 additionally provides a graphical user interface (GUI) that facilitates interaction with the display, sound, and touch and motion input devices.

[0030] Runtime system 320 implements portions of an execution model, including such operations as putting parameters onto the stack before a function call, the behavior of long-term storage input/output (I/O), and parallel execution-related behaviors. Runtime system 320 may also perform support services such as type checking, debugging, or code generation and optimization.

[0031] Libraries 322 include collections of program functions that provide further abstraction for application programs. These include shared libraries, dynamic linked libraries (DLLs), for example. Libraries 322 may be integral to the operating system 318, runtime system 320, or may be added-on features, or even remotely-hosted. Libraries 322 define an application program interface (API) through which a variety of function calls may be made by application programs 324 to invoke the services provided by the operating system 318. Application programs 324, oftentimes referred to as *apps*, are those programs that perform useful tasks for users, beyond the tasks performed by lower-level system programs that coordinate the basis operability of the computing device itself.

[0032] Examples, as described herein, may include, or may operate on, logic or a number of components, modules, or engines, which for the sake of consistency are termed *modules*, although it will be understood that these terms may be used interchangeably. Modules may be hardware, software, or firmware communicatively coupled to one or more processors in order to carry out the operations described herein. Modules may be hardware modules, and as such modules may be considered tangible entities capable of performing specified operations and may be configured or arranged in a certain manner. In an example, circuits may be arranged (e.g., internally or with respect to external entities such as other circuits) in a specified manner as a module.

[0033] In various examples, modules may be formed from circuitry of one or more general-purpose processors, one or more dedicated graphics processors, one or more ASICs, or other hardware circuitry, which may be entirely hard-wired, or may be programmable by program instructions that, when executed on underlying processing hardware, form one or more modules.

[0034] In an example, the whole or part of one or more computer systems (e.g., a standalone, client or server computer system) or one or more hardware processors may be configured by firmware or software (e.g., instructions, an

application portion, or an application) as a module that operates to perform specified operations. In an example, the software may reside on a machine-readable medium. In an example, the software, when executed by the underlying hardware of the module, causes the hardware to perform the specified

5 operations.

[0035] Accordingly, the term hardware module is understood to encompass a tangible entity, be that an entity that is physically constructed, specifically configured (e.g., hardwired), or temporarily (e.g., transitorily) configured (e.g., programmed) to operate in a specified manner or to perform part or all of any operation described herein. With reference to FIG. 3, for instance, a module
10 may include one, or any combination, of the blocks depicted, so long as at least one block from the HW side is included.

[0036] Considering examples in which modules are temporarily configured, each of the modules need not be instantiated at any one moment in time. For example, where the modules comprise a general-purpose hardware processor
15 configured using software; the general-purpose hardware processor may be configured as respective different modules at different times. Software may accordingly configure a hardware processor, for example, to constitute a particular module at one instance of time and to constitute a different module at
20 a different instance of time.

[0037] FIG. 4 is a system block diagram illustrating a system for rendering objects from an image 401 to be displayed on an irregular surface according to some embodiments. Presenter module 402 is configured with hardware and software components, and operative to identify graphical objects to be displayed
25 from the image 401, and to determine if an object to be displayed is to be newly processed to be rendered and deformed, or whether a suitable previously-processed instance of the object may be reused.

[0038] Image 401 may be identified by presenter module 402 as a sprite with an alpha channel. The alpha channel of image 401 maps out the shape of a sprite object. In the absence of an alpha channel, presenter module 402 may identify a
30 graphical object as a raster image and the bounding box of the graphical object may be the rectangular frame of the raster image. For example, a raster image may be added to the rectangular mesh renderer 406 as a background raster image. Next, a text object or sprite may be passed to rectangular mesh renderer

406 and overlaid with the background raster image. Because the graphical object (e.g., the sprite or text) is handled separately from the background part of the image, it may be re-positioned, or removed, by presenter module 402 at a later point in time.

- 5 **[0039]** According to some embodiments, rectangular mesh renderer module 406 is configured with hardware and software components, and operative to computationally represent objects on a rectangular mesh composed of individual deformable elements. Object renderer module 420 is configured with hardware and software components, and operative to apply a forward transformation of an
10 object represented with a rectangular mesh, into a deformed mesh based on one or more shape functions produced by shape function generator 410.

[0040] Object overlay module 408 is configured with hardware and software components, and operative to overlay certain previously-deformed objects to the deformed mesh produced by object renderer module 420.

- 15 **[0041]** Surface characterization module 412 is configured with hardware and software components, and operative to produce a representation of the curvature and other geometric features of an irregular surface (e.g., surface 104), from which shape function generator 410 may generate a series of shape functions.

- [0042]** The shape functions, which are described below in greater detail,
20 implement object deformation by variation of mesh shapes onto which objects are mapped.

- [0043]** Projector driver module 414 is configured with hardware and software components, and operative to produce a video display signal representing the deformed screen image to be displayed onto surface 104 based on the output of
25 object renderer module 420, and object overlay module 408.

[0044] In some embodiments, the graphics rendering for smart glasses 100 is partitioned into the processing of graphics elements on two distinct surfaces: (1) a rectangular mesh of regular elements, called a Lagrange parent elements K_m ; and (2) a deformable mesh of irregular elements, called Lagrange elements D_m .

- 30 The Lagrange parent element K_m is a rectangular rendering element on a surface that is independent of the rendering surface 104 of glasses 100 and fits well into the Cartesian coordinate system. The irregular Lagrange element D_m , in contrast, is dependent on the curvature of the surface 104 of smart glasses

100. This partitioning helps to separate the composition of visual elements/objects (i.e., content) from the pixel interpolation during rendering.

[0045] In related embodiments, the partitioning creates glasses/device independent rendering space for graphics composition so that the routines for graphics manipulation and processing may be portable across different glasses and wearable products. In additional related embodiments, the partitioning supports focusing on performance optimization for the space transform in mapping graphics elements from the regular rectangular space to high-order non-rigid space represented by the deformable mesh. For example, Single-Instruction-Multiple-Data (SIMD) performance optimization may be employed in certain embodiments.

[0046] In a related embodiment, presenter module 402 queries for previously-rendered deformable objects, e.g., a logo at a fixed position, which have been rendered previously and cached in rendered object pool 404.

[0047] FIG. 5 is a diagram illustrating a transformation of a Lagrange parent element to an irregular Lagrange element according to some embodiments. In an example embodiment, the basis element used in the rendering system is a nine-node Lagrange element, as depicted in FIG. 6. In this example, the parent element K_m 502 is a regularly-shaped rectangle, which becomes irregular deformed shape D_m 504 after distortion.

[0048] FIGs. 6 and 7 are a diagrams illustrating a Lagrange parent element, which may be discretized according to some embodiments. In FIG. 6, parent element K_m 502 is a nine-node element. FIG. 7 illustrates a result of

resampling of the width and height of parent element K_m 502 at a sampling interval of 1/15, producing a canvas of size 41 by 41 pixels, as indicated at 702.

[0049] Shape function generator module 410 produces shape functions that describe the deformation for D_m 504. The function within Lagrange element may be interpolated from its nodal values using the shape function. As an illustrative example, the shape function of a nine-node Lagrange element, such as element 502, is given by:

$$\begin{aligned}
N_1 &= \frac{1}{4}(1 - s_1)(1 - s_2) - \frac{1}{2}N_5 - \frac{1}{2}N_8 + \frac{1}{4}N_9 \\
N_2 &= \frac{1}{4}(1 + s_1)(1 - s_2) - \frac{1}{2}N_5 - \frac{1}{2}N_6 + \frac{1}{4}N_9 \\
N_3 &= \frac{1}{4}(1 + s_1)(1 + s_2) - \frac{1}{2}N_6 - \frac{1}{2}N_7 + \frac{1}{4}N_9 \\
N_4 &= \frac{1}{4}(1 - s_1)(1 + s_2) - \frac{1}{2}N_7 - \frac{1}{2}N_8 + \frac{1}{4}N_9 \\
N_5 &= \frac{1}{2}(1 - s_1^2)(1 - s_2) - \frac{1}{2}N_9 \\
N_6 &= \frac{1}{2}(1 + s_1)(1 - s_2^2) - \frac{1}{2}N_9 \\
N_7 &= \frac{1}{2}(1 - s_1^2)(1 + s_2) - \frac{1}{2}N_9 \\
N_8 &= \frac{1}{2}(1 - s_1)(1 - s_2^2) - \frac{1}{2}N_9 \\
N_9 &= (1 - s_1^2)(1 - s_2^2)
\end{aligned}$$

[0050] In the above example equations, variables S_1 and S_2 are intrinsic coordinates in the parent element K_m . The shape function has a value of unity at the node to which it is related. It also has the property that the sum at any point within an element is equal to unity.

[0051] The effectiveness of any particular element type may depend on how well the shape function is capable of representing the local curvature of the surface 104 of smart glasses 100. In some embodiments, shape function generator 410 is capable of representing second-order of curvature so it gives the description of lens curvature in high precision.

[0052] In order to facilitate reuse of previously-rendered visual objects, some embodiments use locality-sensitive hashing (LSH) to reduce the dimensionality of shape function data. For example, the shape function for each mesh element may be reduced to one of the 10 representative sets of shape functions. Each representative shape function is associated with a hash value from 0 to 9, in an example.

[0053] FIG. 8 illustrates an example of a transformation of a textual object according to some embodiments. As indicated at 802, the visual object, which in the present example is the word "SUNNY" having a regular shape, is transformed according to the shape function based on the above equations, into

deformable object 804, which is based on the object's intrinsic shape, as modified by the applied deformation.

[0054] In some embodiments, each deformable object is associated with one shape function only (or a single hash value of shape functions in embodiments using the caching technique with rendered object pool 404). If the distortion of the visual object is too great, the visual object will be broken down into two or more deformable objects. In the present example, the single word "SUNNY" may be broken into a first part, "SU" and a second part, "NNY." In a related embodiment, this problem may be solved with the use of multiple deformable objects to represent the visual object, e.g., "SUNNY" if the extent of distortion where "SUNNY" is rendered cannot be accurately represented by one single shape function. This may be computationally determined by an assessment of degree of distortion, and comparing that assessment against a defined distortion threshold.

[0055] FIG. 9 is a diagram illustrating a process of modeling the visual objects to be displayed using a deformable mesh according to some embodiments. As depicted, mesh 902 is a rectangular mesh composed of a plurality of Lagrange parent elements, denoted by K_m 502. Each parent element K_m 502 is designed to approximate a function over a continuous domain.

[0056] In the example embodiment depicted, deformed mesh 904 of irregular elements D_m 504 is treated as being deformed from the parent element's mesh having regular parent elements K_m 502. The deformation represents the compensation for lens distortion for projected visual objects.

[0057] In a related embodiment, coordinates may be interpolated from the nodal values using the same shape function. For instance, as illustrated in FIG. 9, coordinates x_1 and x_2 within an irregular element D_m 504 are defined by:

$$x_1 = \sum_{i=1}^9 N_i s_1(i) \text{ and } x_2 = \sum_{i=1}^9 N_i s_2(i).$$

[0058] By employing this coordinate mapping function, this embodiment may uniquely identify the coordinates (x_1, x_2) on the lens surface 104 of glasses

100 where the pixel value corresponding to the rectangular parent element at coordinates (S_1, S_2) may be placed.

[0059] In some embodiments, a one-time surface characterization may be performed to compute the shape function 410 for each element in deformable mesh 902. The shape function 410 may be derived by identifying the placement of the nine nodal positions on lens surface 104. The surface characterization operations may be performed at the manufacturer's factory and the shape function 410 may be common for glasses 100 if the same lens and the same projector 102 and projection arrangement (e.g., projector 102 position relative to the lens surface 104) are used.

[0060] In a related embodiment, the shape functions 410 for different display locations are different if the underlying lens curvature varies significantly (e.g., beyond a threshold). For example, if there is more than one location for displaying a screen on the lens surface 104, each location may be associated with a corresponding shape function because the curvatures on different part of lens may vary.

[0061] In some embodiments, object renderer 420 is configured with hardware and software components, and operative to transform a rectangular mesh (e.g., mesh 902) into a deformed mesh (e.g., mesh 904) using the shape functions 410 received from surface characterization module 412.

Deformable Object #	Shape Function Hash	Description	Memory Address
1	4	Logo 'Star'	0x3E13A44D
2	6	Letter 'A'	0x3E12B35E
3	7	Word 'SUNNY'	0x3E22A44A
4	6	Letter 'B'	0x3E12B36D
5	6	Letter 'C'	0x3E12B37E

Table 1

[0062] In order to reduce the overall computational load for this operation, some embodiments reuse previously-rendered deformable objects when matching objects are found in rendered object pool 404. Table 1 represents an example database of rendered object pool 404 according to an embodiment. To

find a matching object, a query of rendered object pool 404 seeks matches of both, the shape function hash value, and a description, of the object of interest.

[0063] For example, with reference to Table 1 above, to carry out the task of rendering the text string “ABC” for display according to a shape function that has a shape function hash value of 6, deformable objects 2, 4 and 5 are found to be matching objects. Instead of computing the deformed objects ABC again, presenter module 402 simply fetches the deformable objects 2, 4 and 5 from the rendered object pool 404, in this case, those objects located at memory addresses 0x3E12B35E, 0x3E12B36D and 0x3E12B37E.

[0064] FIG. 10 is a flow diagram illustrating an example of a process 1000 to correct an image for display on a surface having an irregular geometry according to some embodiments. Example process 1000 may be performed by the system of FIG. 4, or by a system having a different architecture. Notably, process 1000 is a machine-implemented process that operates autonomously (e.g., without user interaction), though the operation of the process may be user initiated, or automatically initiated. In addition, it is important to note that process 1000 is an embodiment that may be realized as described; in addition, portions of the process may be implemented while others are omitted in various embodiments. The following Additional Notes and Examples section details various combinations, without limitation, that are contemplated. It should also be noted that in various embodiments, certain process operations may be performed in a different ordering than depicted in FIG. 10.

[0065] At 1002, image data to be displayed is accessed. In various embodiments, the image prominently features iconic or textual information. At 1004, particular graphical objects are identified. These may be letters, words, icons, sprites, or, more generally, contrasting shapes with identifiable edges or boundaries. In related embodiments, a plurality of distinct objects may be identified from each single image.

[0066] At 1006, based on the location of the graphical object within the display image, a shape function for distorting the object is determined. This determination may be computationally achieved via a lookup operation that reads a table which maps various locations to corresponding shape functions (which have been previously determined in a surface characterization operation. Accordingly, in some embodiments, a deformable object is defined as a

- distortion-compensated graphical object. The deformable object may be derived from the original graphical object by putting the original visual object in the framework of a Lagrange parent element and applying forward transformation via the applicable shape function. This processing operation may be performed
- 5 at operation 1012. Notably, the processing is resource-intensive.
- [0067] In a related embodiment, prior to initiating processing to transform the graphical object into a distorted version of itself, operation 1008 checks if the same object has been previously processed with the determined shape function. According to decision 1010, if a preprocessed object is not available, operation
- 10 1012 performs the shape distortion processing according to the shape function. Otherwise, if the preprocessed object is available, the process branches to operation 1014, which accesses the preprocessed object, and adds it to the output image.
- 15 Additional Notes & Examples:
- [0068] Example 1 is a system for producing an image on an irregular surface, the system comprising: a presenter module to identify a graphical object from an image to be displayed on the irregular surface; an object renderer module to distort objects according to at least one shape function, wherein distortion of the
- 20 objects compensates for irregularities in the irregular surface; and an object overlay module to add previously-distorted instances of objects to a distortion-compensated image; wherein the presenter module is further to determine whether the graphical object is to be processed by the object renderer module prior to being processed by the object overlay module
- 25 [0069] In Example 2, the subject matter of Example 1 optionally includes, further comprising: computing hardware, including a processor, a data store, and input/output facilities, data store containing instructions that, when executed, cause the computing hardware to implement the presenter module, the object renderer module, and the object overlay module.
- 30 [0070] In Example 3, the subject matter of any one or more of Examples 1–2 optionally include, wherein at least one of the presenter module, the object renderer module, and the object overlay module, is implemented in hard-wired logic.

- [0071] In Example 4, the subject matter of any one or more of Examples 1–3 optionally include, further comprising: a rectangular mesh renderer module to computationally represent objects on a rectangular mesh composed of individual deformable elements, wherein the presenter module is to pass the graphical
5 object to the rectangular mesh renderer module.
- [0072] In Example 5, the subject matter of Example 4 optionally includes, wherein the individual deformable elements are Lagrange parent elements.
- [0073] In Example 6, the subject matter of any one or more of Examples 4–5 optionally include, wherein the object renderer module is to transform at least
10 some of the individual deformable elements into irregular Lagrange elements based on application of at least one shape function.
- [0074] In Example 7, the subject matter of any one or more of Examples 1–6 optionally include, further comprising: a rendered object pool that includes a database of the previously-distorted instances of objects, the rendered object
15 pool being responsive to queries identifying original objects and type of distortion applied to those original objects.
- [0075] In Example 8, the subject matter of Example 7 optionally includes, wherein the rendered object pool is to store hashes of shape functions in association with objects to which those shape functions have been applied to
20 distort those objects.
- [0076] In Example 9, the subject matter of any one or more of Examples 1–8 optionally include, wherein the object renderer module is to apply a plurality of different shape functions to the graphical object, the plurality of different shape functions corresponding to a plurality of different positions for the graphical
25 object to be displayed on the irregular surface.
- [0077] In Example 10, the subject matter of any one or more of Examples 1–9 optionally include, wherein the object renderer module is to divide the graphical object into a plurality of graphical objects in response to a degree of applied distortion exceeding a defined threshold.
- [0078] In Example 11, the subject matter of any one or more of Examples 1–10
30 optionally include, wherein the object renderer module is to apply a shape function to the graphical object that is at least a second-order shape function.

[0079] In Example 12, the subject matter of any one or more of Examples 1–11 optionally include, wherein the system further comprises a display projector arranged to project the image onto the irregular surface.

5 [0080] In Example 13, the subject matter of any one or more of Examples 1–12 optionally include, wherein the system is incorporated as part of a wearable computing device.

[0081] In Example 14, the subject matter of any one or more of Examples 1–13 optionally include, wherein the system is incorporated as part of a smartphone.

10 [0082] In Example 15, the subject matter of any one or more of Examples 1–14 optionally include, wherein the system is operative to be communicatively coupled to a remote computing device that includes a display projector arranged to project the image onto the irregular surface.

[0083] Example 16 is a method for producing an image on an irregular surface, the method comprising: identifying, by a computing device, a graphical object
15 from a first image to be displayed on the irregular surface; determining, by a computing device, a shape function for distorting the graphical object to compensate for irregularities in the irregular display surface; processing, by a computing device, a first distortion of the graphical object according to the shape function if a preprocessed instance of the first distortion is not available;
20 accessing, by a computing device, a preprocessed instance of the first distortion if the preprocessed instance is available; and overlaying, by a computing device, the first distortion to an output image for display on the irregular surface

[0084] In Example 17, the subject matter of Example 16 optionally includes, wherein processing the first distortion includes computationally representing the
25 graphical object on a rectangular mesh composed of individual deformable elements.

[0085] In Example 18, the subject matter of Example 17 optionally includes, wherein the individual deformable elements are Lagrange parent elements.

30 [0086] In Example 19, the subject matter of any one or more of Examples 17–18 optionally include, wherein processing the first distortion includes transforming at least some of the individual deformable elements into irregular Lagrange elements based on application of the shape function.

[0087] In Example 20, the subject matter of any one or more of Examples 16–19 optionally include, wherein shape function is at least a second-order shape function.

5 [0088] In Example 21, the subject matter of any one or more of Examples 16–20 optionally include, further comprising: storing a database of preprocessed instances of objects, including the first distortion.

[0089] In Example 22, the subject matter of any one or more of Examples 16–21 optionally include, further comprising: determining a plurality of different additional shape functions for distorting the graphical object according to
10 different additional positions for the graphical object to be displayed on the irregular surface.

[0090] In Example 23, the subject matter of any one or more of Examples 16–22 optionally include, further comprising: dividing the graphical object into a plurality of graphical objects in response to a degree of applied distortion
15 exceeding a defined threshold.

[0091] In Example 24, the subject matter of any one or more of Examples 16–23 optionally include, further comprising: projecting the output image onto the irregular surface.

[0092] In Example 25, the subject matter of any one or more of Examples 16–20
20 24 optionally include, wherein the method is executed automatically by a wearable computing device.

[0093] In Example 26, the subject matter of any one or more of Examples 16–25 optionally include, wherein the method is executed by a smartphone.

[0094] Example 27 is at least one computer-readable medium comprising
25 instructions that, when executed on computing hardware, cause the computing hardware to implement: a presenter module to identify a graphical object from an image to be displayed on the irregular surface; an object renderer module to distort objects according to at least one shape function, wherein distortion of the objects compensates for irregularities in the irregular surface; and an object
30 overlay module to add previously-distorted instances of objects to a distortion-compensated image; wherein the presenter module is further to determine whether the graphical object is to be processed by the object renderer module prior to being processed by the object overlay module

- [0095] In Example 28, the subject matter of Example 27 optionally includes, wherein the instructions are further to cause the computing hardware to implement: a rectangular mesh renderer module to computationally represent objects on a rectangular mesh composed of individual deformable elements, wherein the presenter module is to pass the graphical object to the rectangular mesh renderer module.
- [0096] In Example 29, the subject matter of any one or more of Examples 27–28 optionally include, wherein the individual deformable elements are Lagrange parent elements.
- 10 [0097] In Example 30, the subject matter of Example 29 optionally includes, wherein the object renderer module is to transform at least some of the individual deformable elements into irregular Lagrange elements based on application of at least one shape function.
- [0098] In Example 31, the subject matter of any one or more of Examples 27–15 30 optionally include, wherein the instructions are further to cause the computing hardware to implement: a rendered object pool that includes a database of the previously-distorted instances of objects, the rendered object pool being responsive to queries identifying original objects and type of distortion applied to those original objects.
- 20 [0099] In Example 32, the subject matter of Example 31 optionally includes, wherein the rendered object pool is to store hashes of shape functions in association with objects to which those shape functions have been applied to distort those objects.
- [00100] In Example 33, the subject matter of any one or more of Examples 27–25 32 optionally include, wherein the object renderer module is to apply a plurality of different shape functions to the graphical object, the plurality of different shape functions corresponding to a plurality of different positions for the graphical object to be displayed on the irregular surface.
- [00101] In Example 34, the subject matter of any one or more of Examples 27–30 33 optionally include, wherein the object renderer module is to divide the graphical object into a plurality of graphical objects in response to a degree of applied distortion exceeding a defined threshold.

[00102] In Example 35, the subject matter of any one or more of Examples 27–34 optionally include, wherein the object renderer module is to apply a shape function to the graphical object that is at least a second-order shape function.

5 [00103] In Example 36, the subject matter of any one or more of Examples 27–35 optionally include, wherein the at least one computer-readable medium is incorporated as part of a wearable computing device.

[00104] In Example 37, the subject matter of any one or more of Examples 27–36 optionally include, wherein the at least one computer-readable medium is incorporated as part of a smartphone.

10 [00105] Example 38 is a system for producing an image on an irregular surface, the system comprising: means for identifying a graphical object from a first image to be displayed on the irregular surface; means for determining a shape function for distorting the graphical object to compensate for irregularities in the irregular display surface; means for processing a first distortion of the graphical
15 object according to the shape function if a preprocessed instance of the first distortion is not available; means for accessing a preprocessed instance of the first distortion if the preprocessed instance is available; and means for overlaying the first distortion to an output image for display on the irregular surface

[00106] In Example 39, the subject matter of Example 38 optionally includes,
20 wherein the means for processing the first distortion includes means for computationally representing the graphical object on a rectangular mesh composed of individual deformable elements.

[00107] In Example 40, the subject matter of Example 39 optionally includes, wherein the individual deformable elements are Lagrange parent elements.

25 [00108] In Example 41, the subject matter of any one or more of Examples 39–40 optionally include, wherein the means for processing the first distortion includes means for transforming at least some of the individual deformable elements into irregular Lagrange elements based on application of the shape function.

30 [00109] In Example 42, the subject matter of any one or more of Examples 38–41 optionally include, wherein shape function is at least a second-order shape function.

[00110] In Example 43, the subject matter of any one or more of Examples 38–42 optionally include, further comprising: means for storing a database of preprocessed instances of objects, including the first distortion.

5 [00111] In Example 44, the subject matter of any one or more of Examples 38–43 optionally include, further comprising: means for determining a plurality of different additional shape functions for distorting the graphical object according to different additional positions for the graphical object to be displayed on the irregular surface.

10 [00112] In Example 45, the subject matter of any one or more of Examples 38–44 optionally include, further comprising: means for dividing the graphical object into a plurality of graphical objects in response to a degree of applied distortion exceeding a defined threshold.

15 [00113] In Example 46, the subject matter of any one or more of Examples 38–45 optionally include, further comprising: the means for projecting the output image onto the irregular surface.

[00114] In Example 47, the subject matter of any one or more of Examples 38–46 optionally include, wherein the system is part of a wearable computing device.

20 [00115] In Example 48, the subject matter of any one or more of Examples 38–47 optionally include, wherein the system is part of a smartphone.

[00116] In Example 49, a system for producing an image on an irregular surface includes means for performing the subject matter according to any one of Examples 16-26.

25 [00117] In Example 50, at least one computer-readable medium contains instructions that, when executed on a computing system, cause the computing system to execute the subject matter according to any one of Examples 16-26.

[00118] The above detailed description includes references to the accompanying drawings, which form a part of the detailed description. The drawings show, by way of illustration, specific embodiments that may be practiced. These
30 embodiments are also referred to herein as “examples.” Such examples may include elements in addition to those shown or described. However, also contemplated are examples that include the elements shown or described. Moreover, also contemplated are examples using any combination or permutation of those elements shown or described (or one or more aspects

thereof), either with respect to a particular example (or one or more aspects thereof), or with respect to other examples (or one or more aspects thereof) shown or described herein.

5 [00119] Publications, patents, and patent documents referred to in this document are incorporated by reference herein in their entirety, as though individually incorporated by reference. In the event of inconsistent usages between this document and those documents so incorporated by reference, the usage in the incorporated reference(s) are supplementary to that of this document; for irreconcilable inconsistencies, the usage in this document controls.

10 [00120] In this document, the terms “a” or “an” are used, as is common in patent documents, to include one or more than one, independent of any other instances or usages of “at least one” or “one or more.” In this document, the term “or” is used to refer to a nonexclusive or, such that “A or B” includes “A but not B,” “B but not A,” and “A and B,” unless otherwise indicated. In the appended
15 claims, the terms “including” and “in which” are used as the plain-English equivalents of the respective terms “comprising” and “wherein.” Also, in the following claims, the terms “including” and “comprising” are open-ended, that is, a system, device, article, or process that includes elements in addition to those listed after such a term in a claim are still deemed to fall within the scope of that
20 claim. Moreover, in the following claims, the terms “first,” “second,” and “third,” etc. are used merely as labels, and are not intended to suggest a numerical order for their objects.

[00121] The above description is intended to be illustrative, and not restrictive. For example, the above-described examples (or one or more aspects thereof)
25 may be used in combination with others. Other embodiments may be used, such as by one of ordinary skill in the art upon reviewing the above description. The Abstract is to allow the reader to quickly ascertain the nature of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. Also, in the above Detailed
30 Description, various features may be grouped together to streamline the disclosure. However, the claims may not set forth every feature disclosed herein as embodiments may feature a subset of said features. Further, embodiments may include fewer features than those disclosed in a particular example. Thus, the following claims are hereby incorporated into the Detailed Description, with

a claim standing on its own as a separate embodiment. The scope of the embodiments disclosed herein is to be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled.

CLAIMS

What is claimed is:

1. A system for producing an image on an irregular surface, the system

5 comprising:

a presenter module to identify a graphical object from an image to be displayed on the irregular surface;

an object renderer module to distort objects according to at least one shape function, wherein distortion of the objects compensates for irregularities in the irregular surface; and

10 an object overlay module to add previously-distorted instances of objects to a distortion-compensated image;

wherein the presenter module is further to determine whether the graphical object is to be processed by the object renderer module prior to being processed by the object overlay module.

2. The system of claim 1, further comprising:

computing hardware, including a processor, a data store, and input/output facilities, data store containing instructions that, when executed, cause the computing hardware to implement the presenter module, the object

3. The system of claim 1, wherein at least one of the presenter module, the object renderer module, and the object overlay module, is implemented in hard-wired logic.

4. The system of claim 1, further comprising:

a rectangular mesh renderer module to computationally represent objects on a rectangular mesh composed of individual deformable elements, wherein the presenter module is to pass the graphical object to the rectangular mesh renderer module.

5. The system of claim 4, wherein the individual deformable elements are Lagrange parent elements.

6. The system of claim 4, wherein the object renderer module is to transform at least some of the individual deformable elements into irregular Lagrange elements based on application of at least one shape function.

5 7. The system of claim 1, further comprising:
a rendered object pool that includes a database of the previously-distorted instances of objects, the rendered object pool being responsive to queries identifying original objects and type of distortion applied to those original objects.

10 8. The system of claim 7, wherein the rendered object pool is to store hashes of shape functions in association with objects to which those shape functions have been applied to distort those objects.

9. The system of claim 1, wherein the object renderer module is to apply a
15 plurality of different shape functions to the graphical object, the plurality of different shape functions corresponding to a plurality of different positions for the graphical object to be displayed on the irregular surface.

10. The system of claim 1, wherein the object renderer module is to divide the
20 graphical object into a plurality of graphical objects in response to a degree of applied distortion exceeding a defined threshold.

11. The system of claim 1, wherein the object renderer module is to apply a shape
function to the graphical object that is at least a second-order shape function.

25 12. The system of claim 1, wherein the system further comprises a display projector arranged to project the image onto the irregular surface.

13. The system of claim 1, wherein the system is incorporated as part of a
30 wearable computing device.

14. The system of claim 1, wherein the system is incorporated as part of a smartphone.

15. The system of claim 1, wherein the system is operative to be communicatively coupled to a remote computing device that includes a display projector arranged to project the image onto the irregular surface.

5 16. A method for producing an image on an irregular surface, the method comprising:

identifying, by a computing device, a graphical object from a first image to be displayed on the irregular surface;

10 determining, by a computing device, a shape function for distorting the graphical object to compensate for irregularities in the irregular display surface;

processing, by a computing device, a first distortion of the graphical object according to the shape function if a preprocessed instance of the first distortion is not available;

15 accessing, by a computing device, a preprocessed instance of the first distortion if the preprocessed instance is available; and

overlaying, by a computing device, the first distortion to an output image for display on the irregular surface.

20 17. The method of claim 16, wherein processing the first distortion includes computationally representing the graphical object on a rectangular mesh composed of individual deformable elements.

25 18. The method of claim 17, wherein the individual deformable elements are Lagrange parent elements.

19. The method of claim 17, wherein processing the first distortion includes transforming at least some of the individual deformable elements into irregular Lagrange elements based on application of the shape function.

30 20. The method of claim 16, wherein shape function is at least a second-order shape function.

21. The method of claim 16, further comprising:

storing a database of preprocessed instances of objects, including the first distortion.

22. The method of claim 16, further comprising:

5 determining a plurality of different additional shape functions for distorting the graphical object according to different additional positions for the graphical object to be displayed on the irregular surface.

23. The method of claim 16, further comprising:

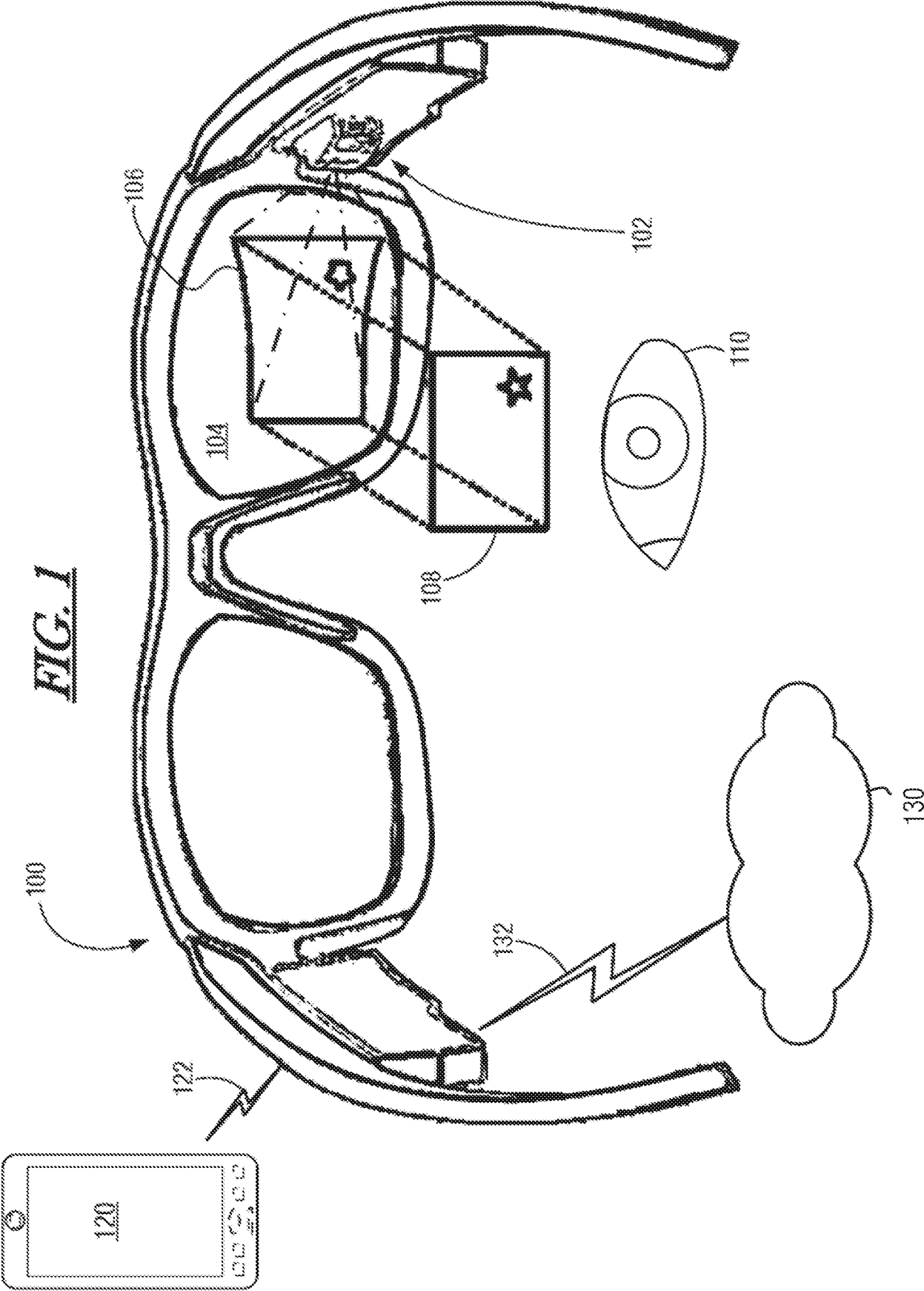
10 dividing the graphical object into a plurality of graphical objects in response to a degree of applied distortion exceeding a defined threshold.

24. The method of claim 16, further comprising:

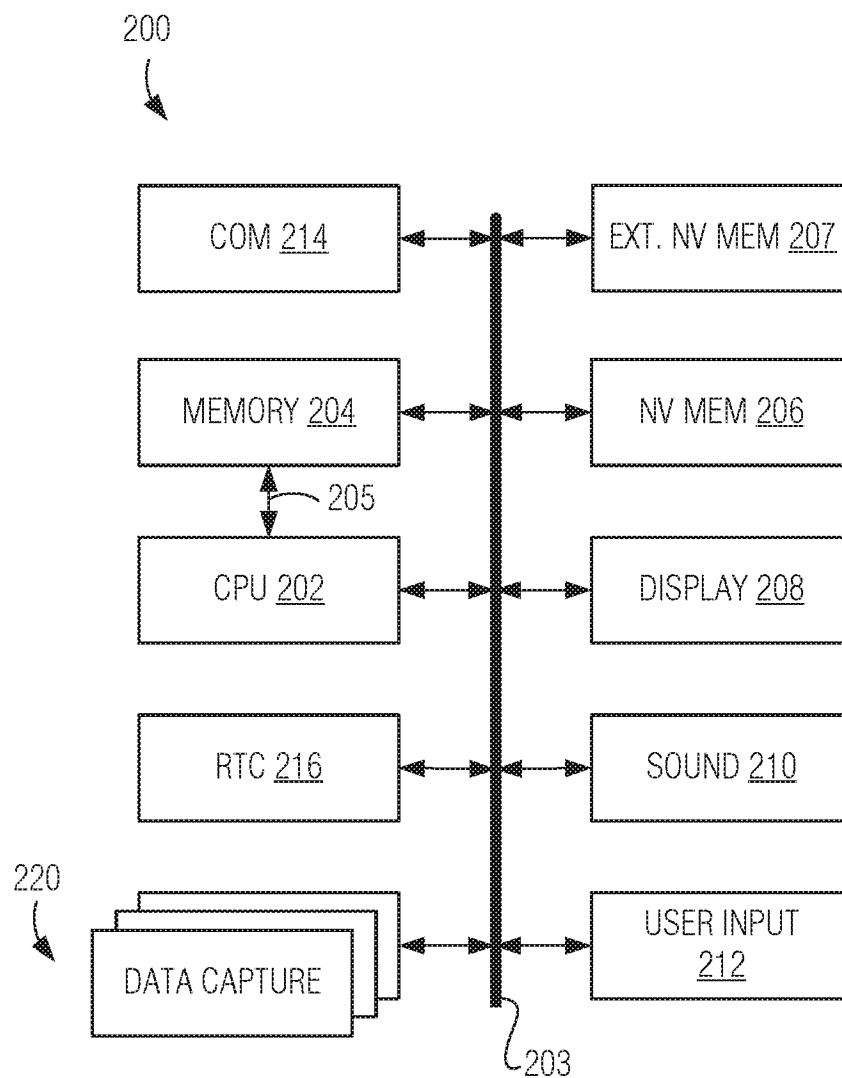
15 projecting the output image onto the irregular surface.

25. A system for producing an image on an irregular surface, the system comprising means for performing the method according to any one of claims 16-24.

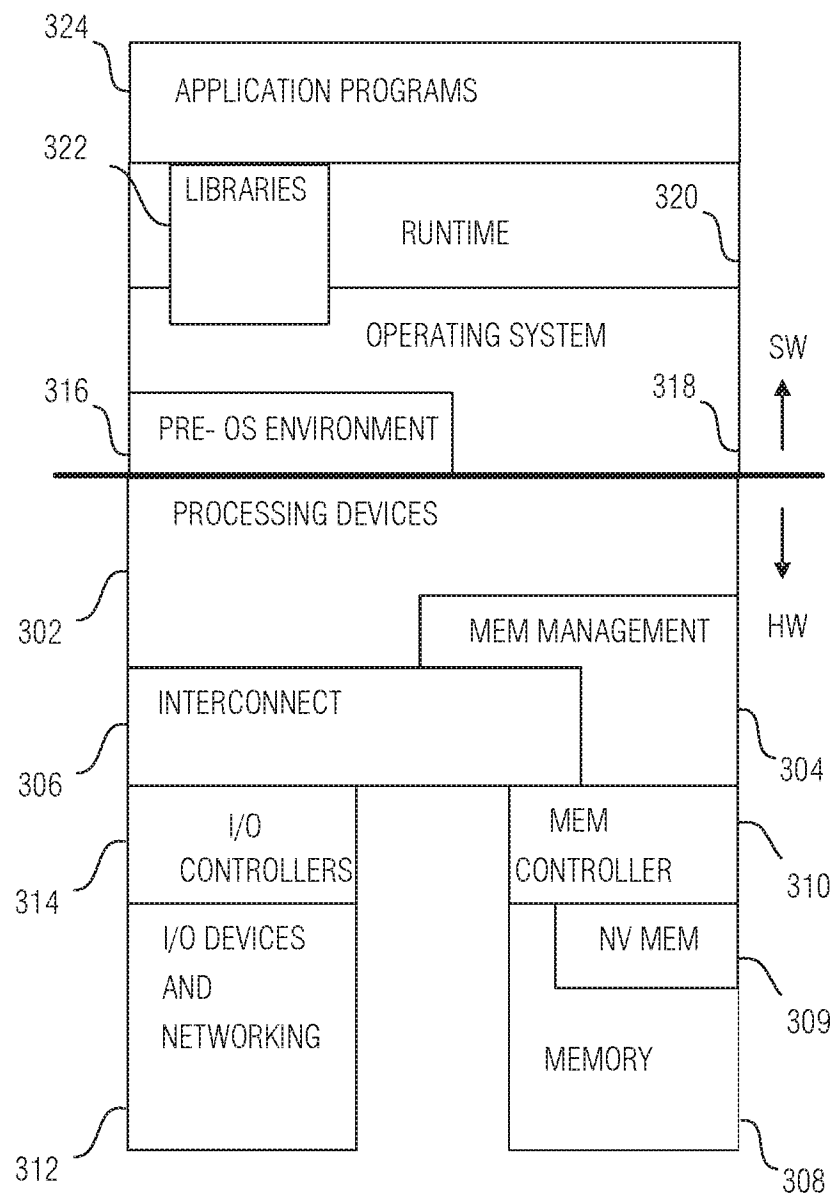
20 26. At least one computer-readable medium comprising instructions that, when executed on a computing system, cause the computing system to execute the method according to any one of claims 16-24.



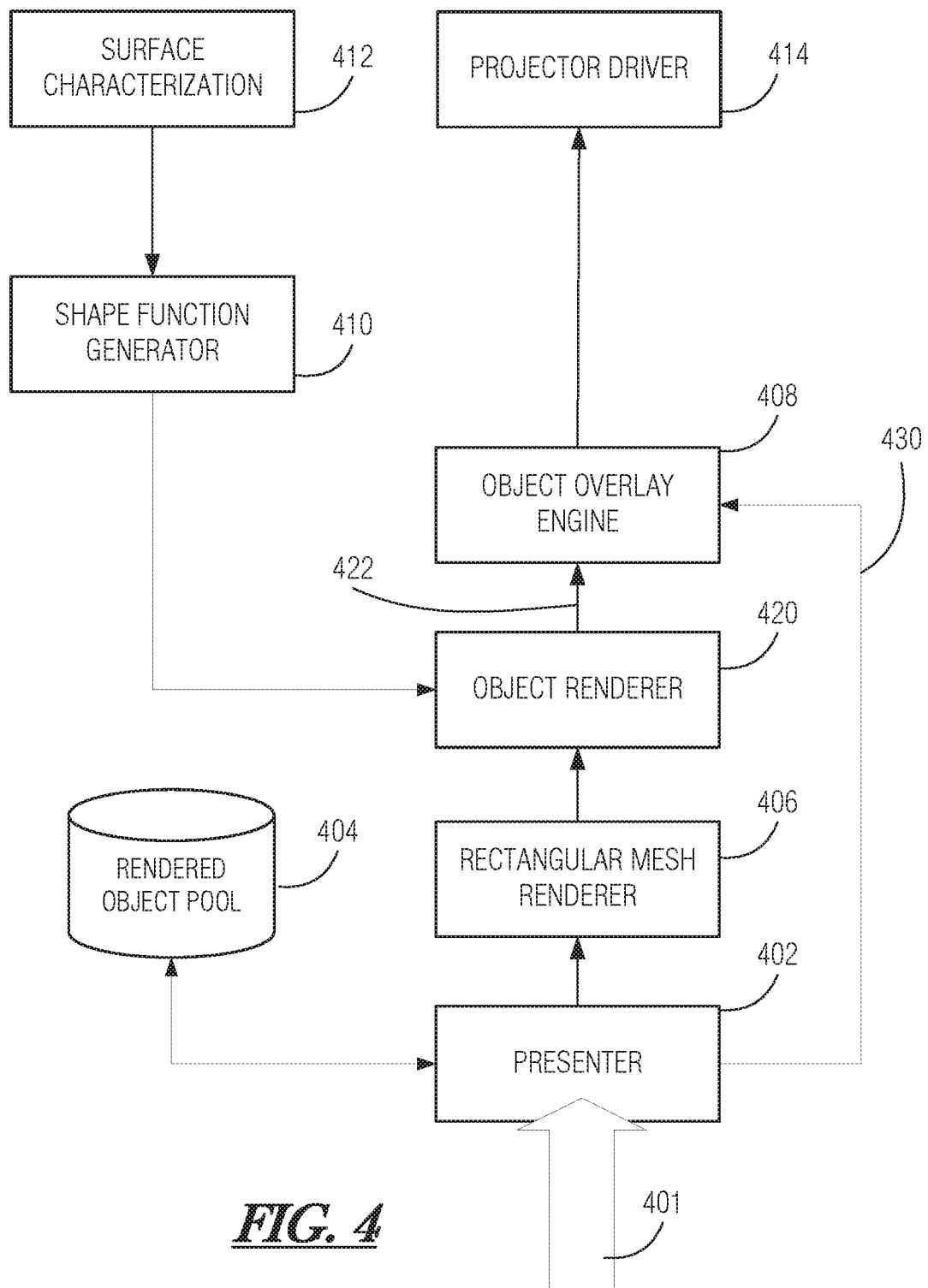
2/10

***FIG. 2***

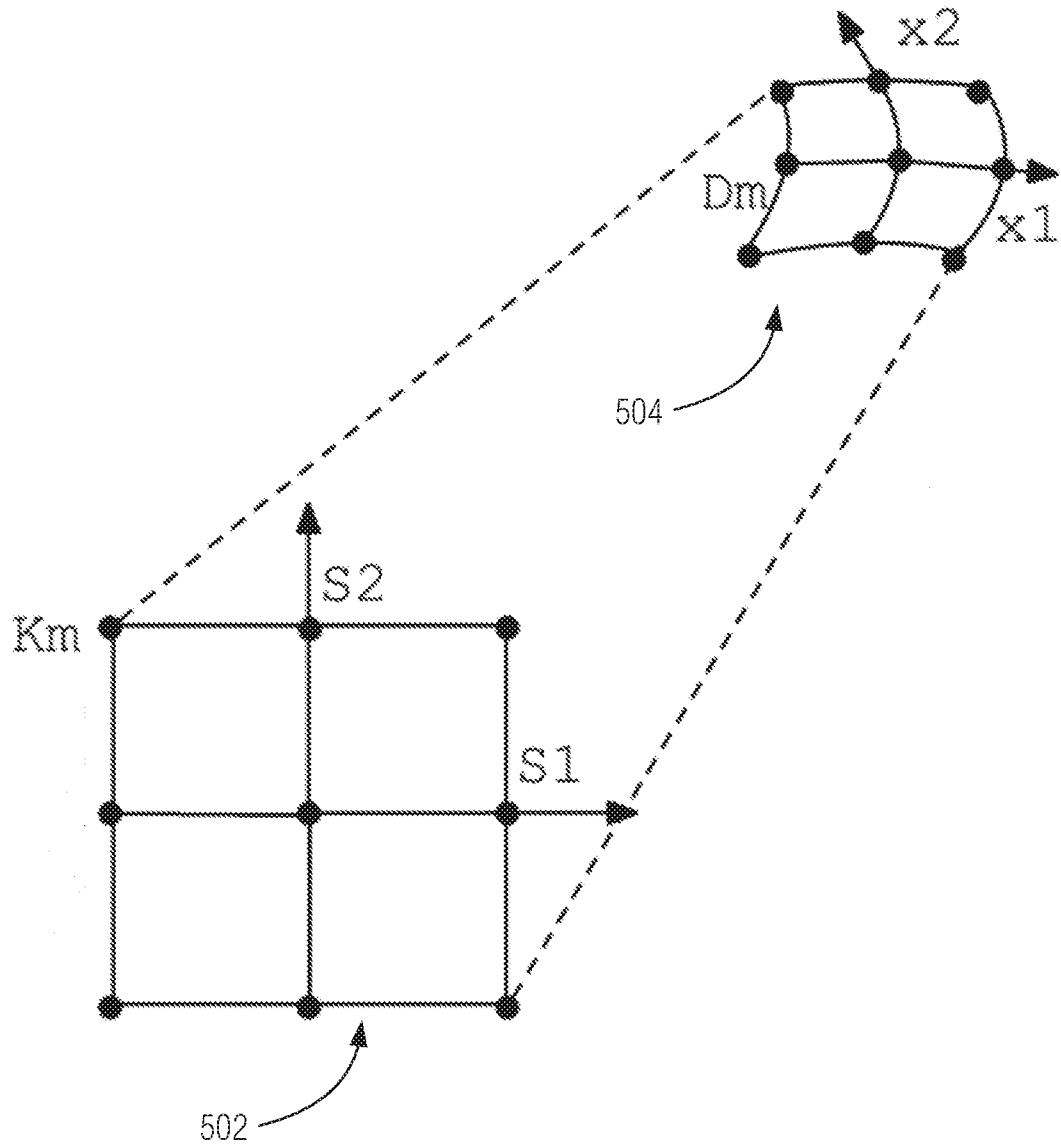
3/10

***FIG. 3***

4/10



5/10

**FIG. 5**

6/10

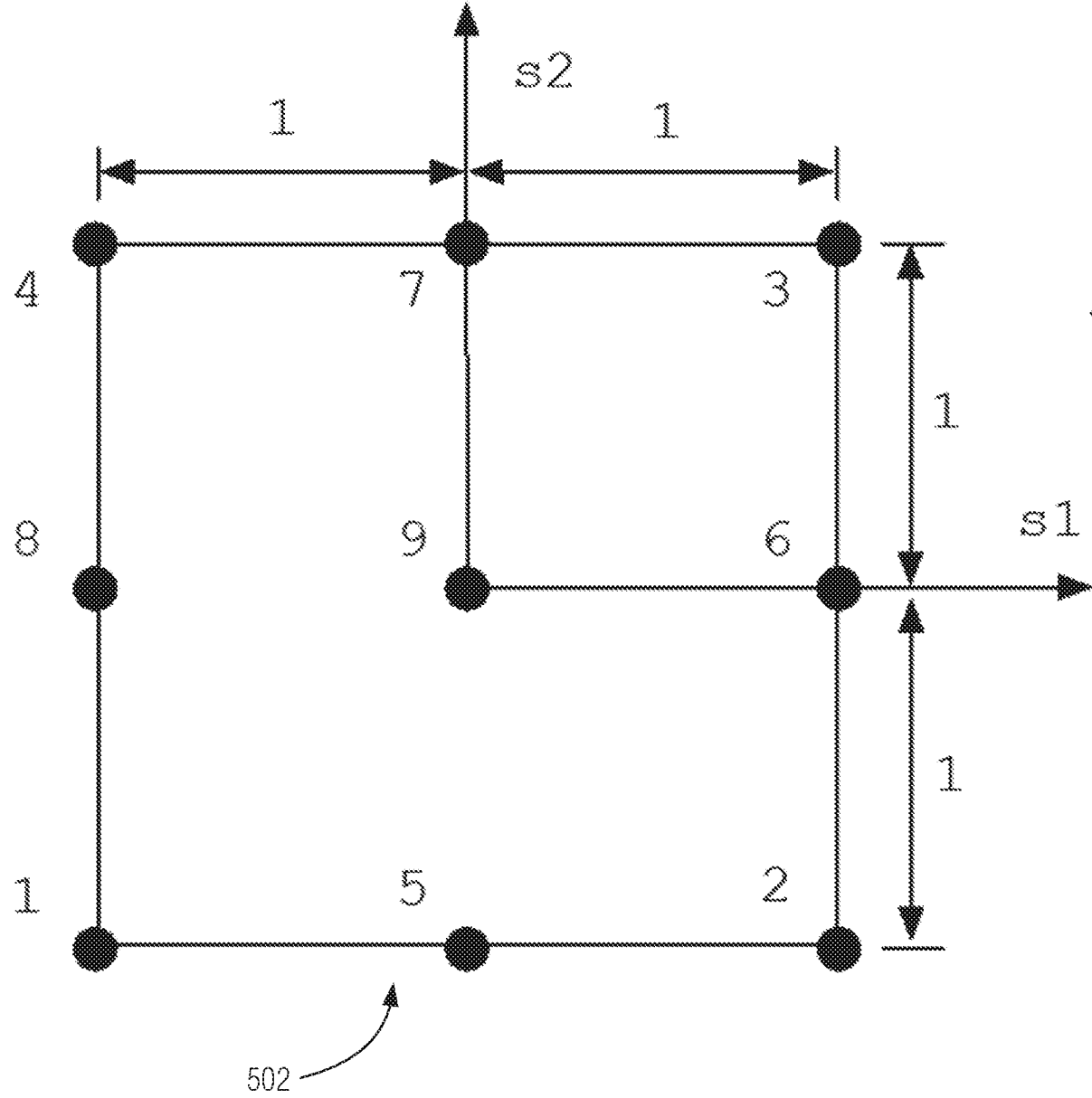


FIG. 6

7/10

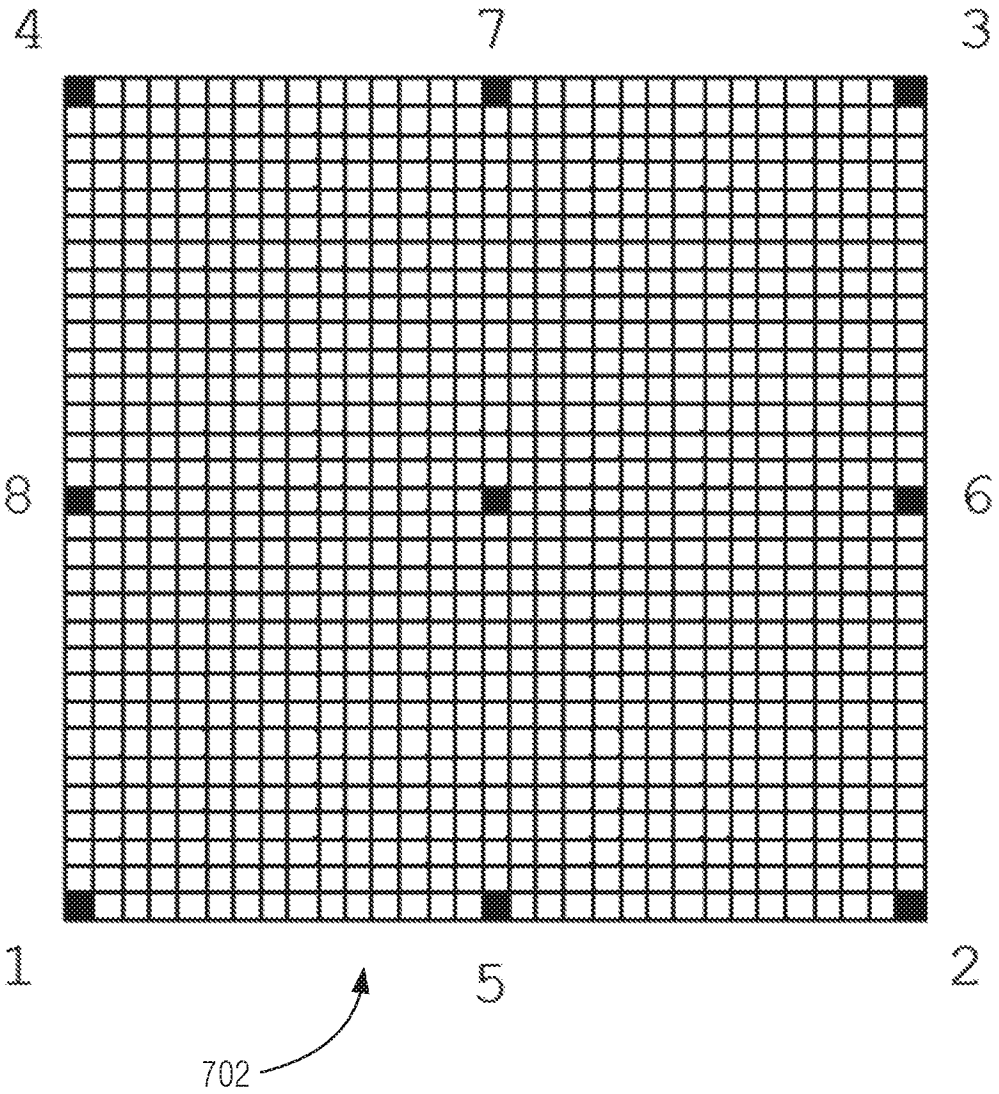


FIG. 7

8/10

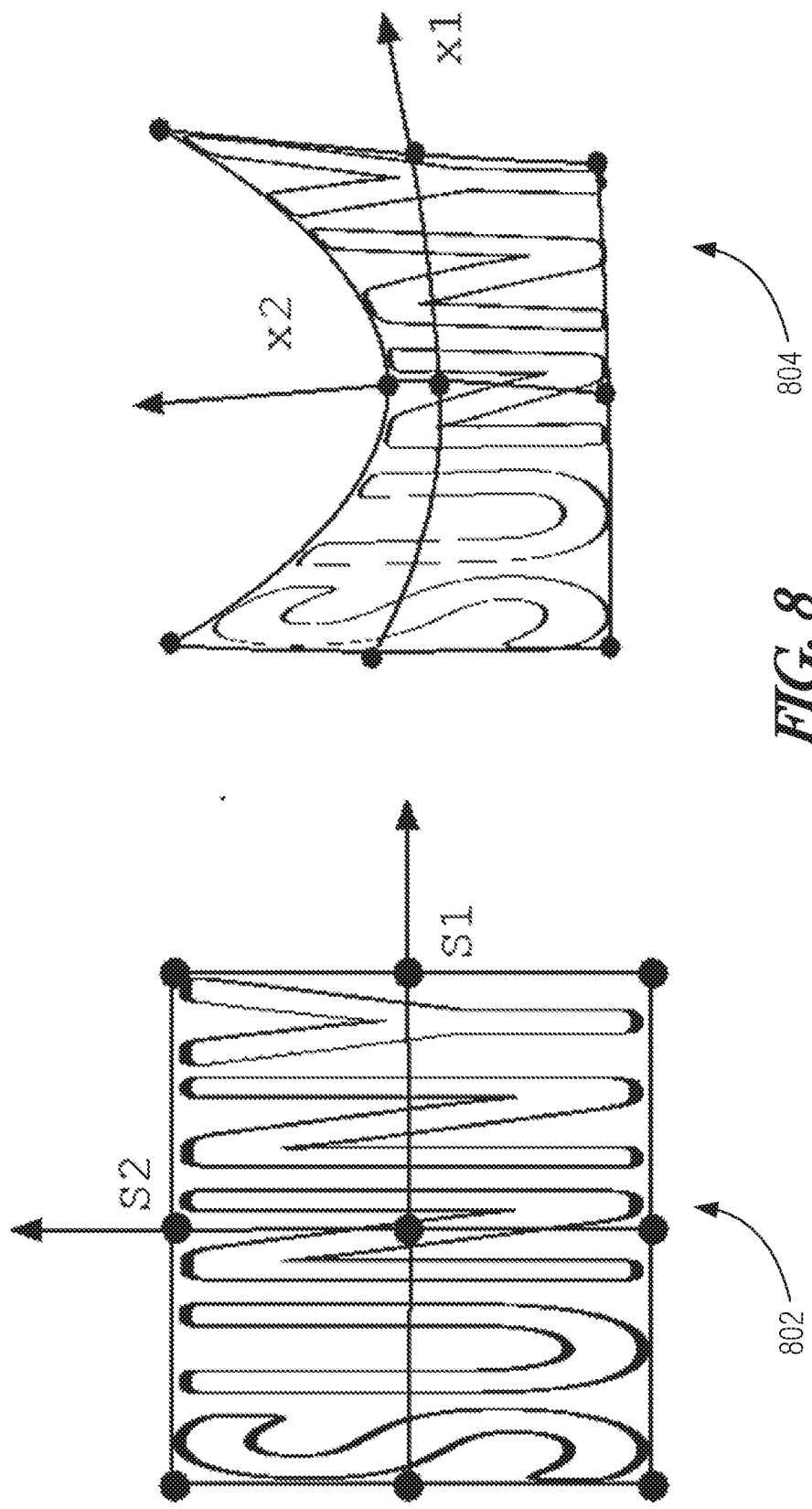
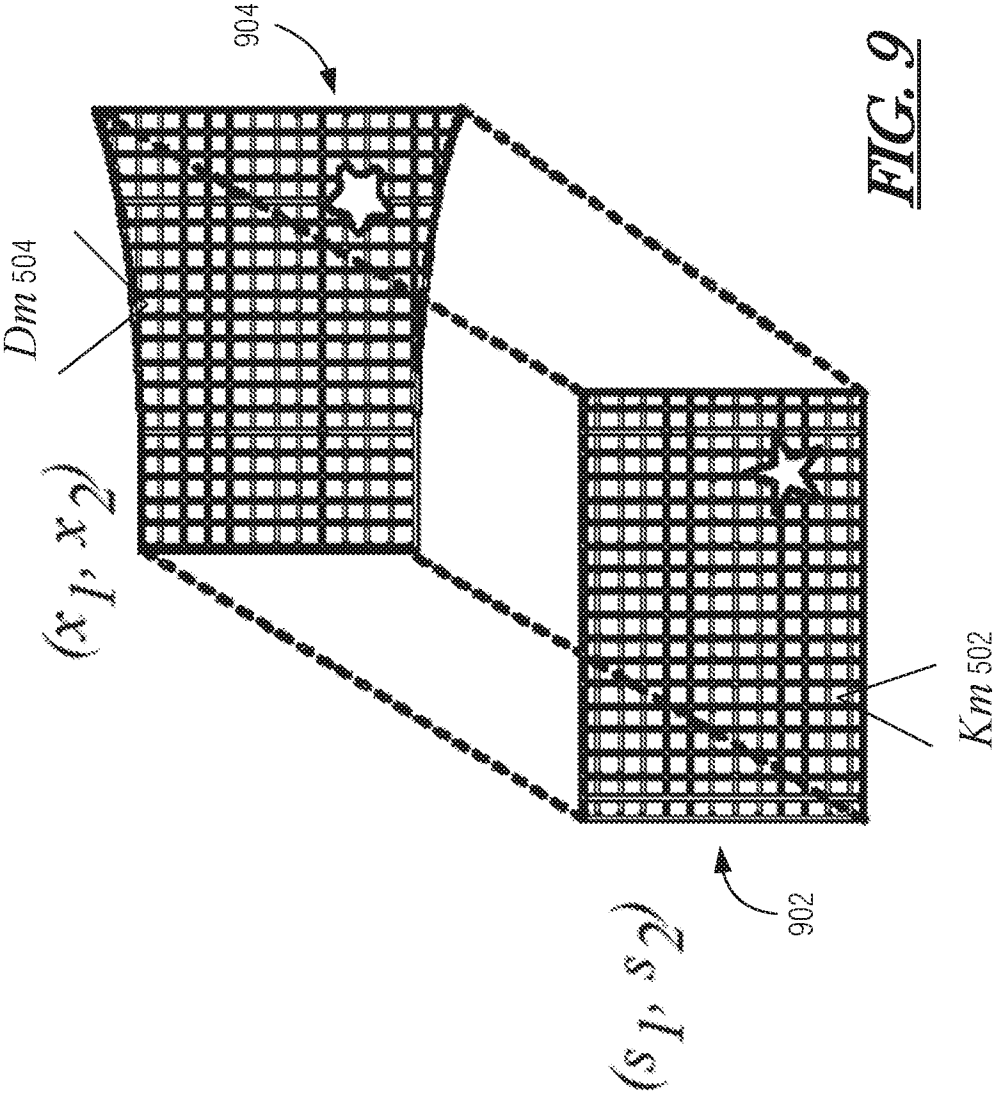
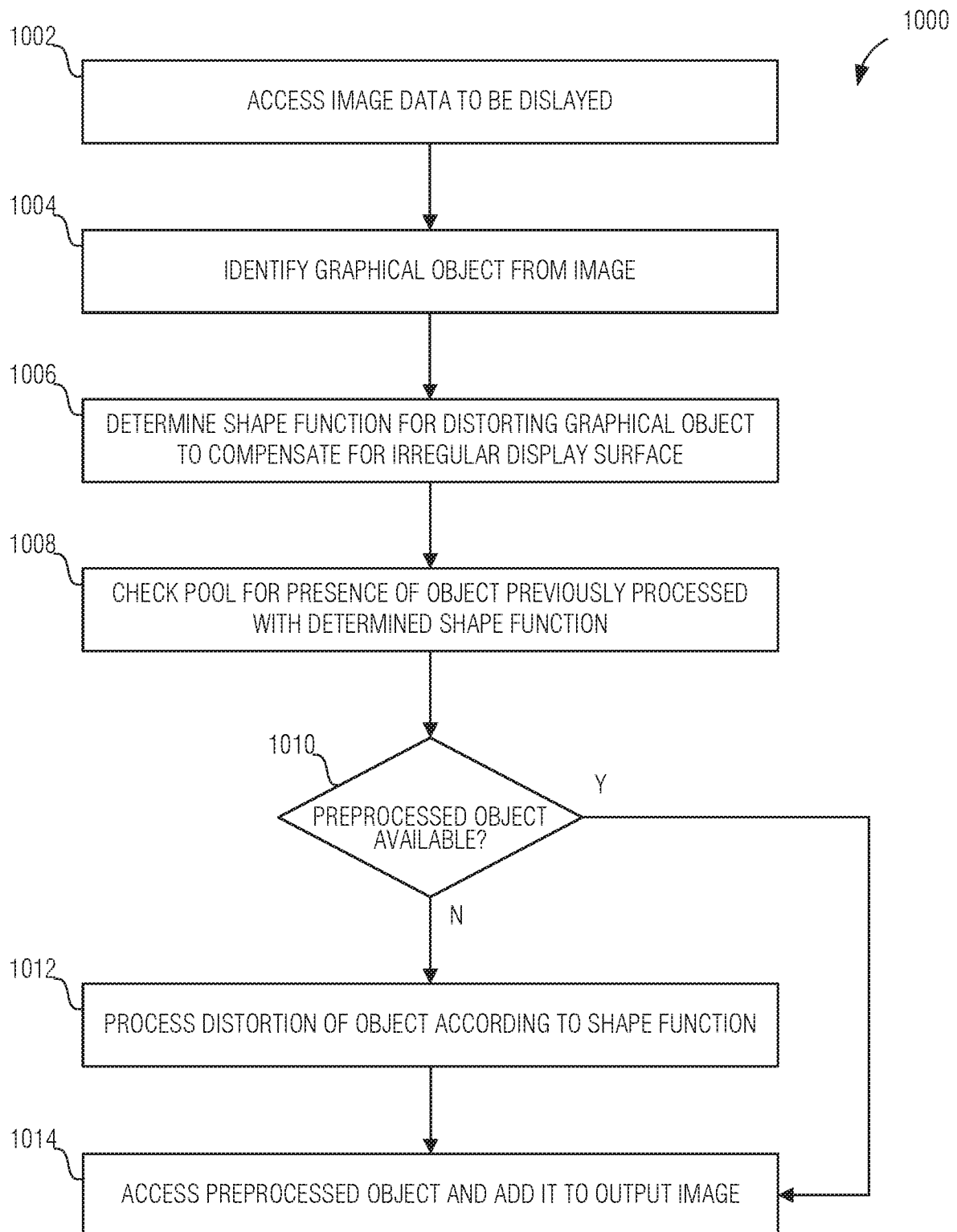


FIG. 8



10/10

***FIG. 10***

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2016/061376**A. CLASSIFICATION OF SUBJECT MATTER****G06T 3/00(2006.01)i, G06T 5/00(2006.01)i, G06T 3/40(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06T 3/00; G09G 5/02; G06T 17/00; G01B 11/24; H04N 13/02; H04N 5/74; G01B 11/25; G06T 5/00; G06T 3/40

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords:image distortion, irregular surface, compensate, smart glasses, graphic object

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2014-0204083 A1 (BRENT THOMSON) 24 July 2014 See paragraphs [0041]-[0045]; claims 1, 2, 13-15; and figures 1, 5.	1-26
A	US 2005-0134599 A1 (SHREE NAYAR et al.) 23 June 2005 See paragraphs [0126]-[0128]; claim 43; and figures 1-3, 13.	1-26
A	US 2010-0245684 A1 (JING XIAO et al.) 30 September 2010 See paragraphs [0118]-[0142]; and figures 1A-3A.	1-26
A	US 2014-0168368 A1 (INSTITUT NATIONAL D'OPTIQUE) 19 June 2014 See paragraphs [0049]-[0069]; and figures 3A-7.	1-26
A	US 2015-0153164 A1 (8TREE, LLC) 04 June 2015 See paragraphs [0032]-[0039]; and figures 5-6.	1-26



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

15 February 2017 (15.02.2017)

Date of mailing of the international search report

20 February 2017 (20.02.2017)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

AHN, Jeong Hwan

Telephone No. +82-42-481-8633



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/061376

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014-0204083 A1	24/07/2014	None	
US 2005-0134599 A1	23/06/2005	US 7663640 B2 WO 2005-015490 A2 WO 2005-015490 A3 WO 2005-015490 A8	16/02/2010 17/02/2005 24/08/2006 22/02/2007
US 2010-0245684 A1	30/09/2010	US 7901095 B2	08/03/2011
US 2014-0168368 A1	19/06/2014	CA 2799705 A1 CA 2799705 C US 9131219 B2	18/06/2014 11/10/2016 08/09/2015
US 2015-0153164 A1	04/06/2015	CN 104487800 A CN 104487800 B EP 2875311 A1 HK 1208902 A1 JP 2015-523574 A US 2014-015963 A1 US 2016-123893 A1 US 8937657 B2 WO 2014-014786 A1	01/04/2015 24/08/2016 27/05/2015 18/03/2016 13/08/2015 16/01/2014 05/05/2016 20/01/2015 23/01/2014