



- (51) International Patent Classification:
H04N 13/04 (2006.01) *G06T 19/00* (2011.01)
- (21) International Application Number:
PCT/GB2012/000334
- (22) International Filing Date:
11 April 2012 (11.04.2012)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
1106111.6 11 April 2011 (11.04.2011) GB
1114032.4 15 August 2011 (15.08.2011) GB
- (71) Applicant (for all designated States except US): **NEWS PLUS MEDIA TECHNOLOGIES LTD** [GB/GB]; c/o Mill Asset Management Group Limited, 3 Tenterden Street, London W1S 1TD (GB).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **McCORMICK, Malcolm** [GB/GB]; 125 Knowle Lane, Ecclesall, Sheffield, South Yorkshire S11 9SN (GB). **FORMAN, Mat-**

thew Charles [GB/GB]; 125 Knowle Lane, Ecclesall, Sheffield, South Yorkshire S11 9SN (GB).

(74) Agents: **COZENS, Paul, Denis** et al.; Mathys & Squire LLP, 120 Holborn, London EC1N 2SQ (GB).

(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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, 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,

[Continued on next page]

(54) Title: THREE-DIMENSIONAL DISPLAY SYSTEM

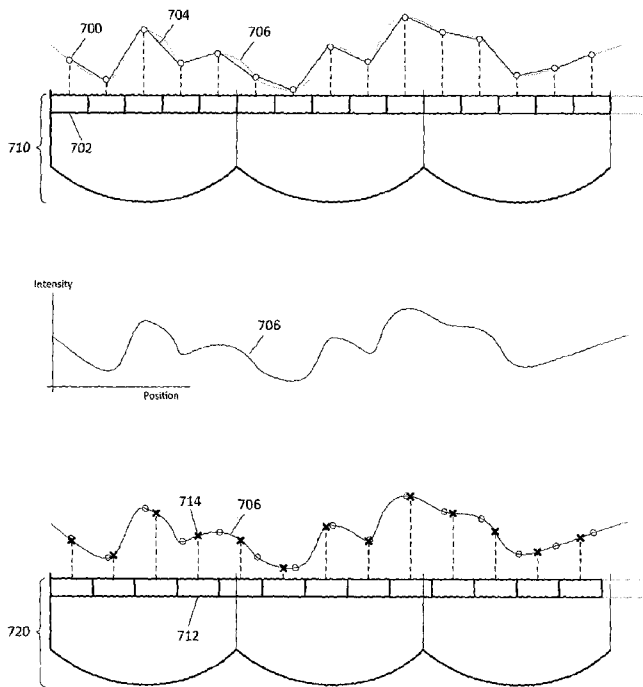


Figure 7

(57) Abstract: The present invention relates to methods, and associated apparatus, for rendering and displaying images in three-dimensional display systems, including autostereoscopic displays. Said method comprises generating a pixel value for each pixel of an image, each pixel being mapped to a single viewpoint, such that an integer number of pixel values are generated for at least one element of a decoder array, storing each generated pixel value, resampling the stored pixel values in dependence on the number of display pixels for each element of the decoder array, and displaying the resampled pixel values. The invention further relates to rendering three-dimensional images for printing, and/or displaying on an electronic display. The invention further relates to a method of adjusting the centre of parallax of an autostereoscopic display. Another aspect of the invention relates to a method, and associated apparatus and system, for providing a display of a three-dimensional scene. Said method comprises receiving an input representative of a given viewpoint, and in dependence on the input, displaying an image providing a view of the three-dimensional scene corresponding to the given viewpoint; the displaying comprises accessing data defining a viewpoint distribution representing a plurality of views of the three-dimensional scene, sampling the viewpoint distribution in dependence on the input to generate a view of the three-dimensional scene corresponding to the given viewpoint, and displaying an image based on the generated view. The invention further relates to a method of, and associated system for, generating a multi-viewpoint representation of a three-dimensional scene, as well as to an associated non-transitory computer-readable medium encoding a data structure.



SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG). **Published:**

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

Three-Dimensional Display System

The invention relates to methods, and associated apparatus, for rendering and displaying images in three-dimensional display systems, including autostereoscopic displays. In particular, the invention relates to rendering three-dimensional images for printing, and/or displaying on an electronic display. The invention further relates to a method of adjusting the centre of parallax of an autostereoscopic display. Another aspect of the invention relates to a method, and associated apparatus and system, for providing a display of a three-dimensional scene. The invention further relates to a method, and associated system, of generating a multi-viewpoint representation of a three-dimensional scene, as well as to an associated non-transitory computer-readable medium encoding a data structure.

For visualising three-dimensional (3D) information a variety of display techniques exist. Different three-dimensional (3D) display techniques rely on stimulating different groups of depth cues within the human visual system. One way to create a 3D effect is by presenting different views of a 3D scene to the viewer's left and right eyes, stimulating the binocular disparity depth cue to produce a stereoscopic display. Stereoscopic display provides the perception of depth, while the display remains two-dimensional. Perception of depth is achieved by combining visual information for two or more viewpoints. This requires a means of ensuring that visual information associated with a particular viewpoint is only seen from that viewpoint. Autostereoscopic displays function without the requirement of glasses or head gear for the viewer. Instead, the display includes a means of decoding the combined visual information. In autostereoscopic displays, the views are presented simultaneously, typically in interlaced form, and a decoding element such as a lenticular array ensures that different images with the correct spatial relationships are seen by each eye. Different views of the 3D scene may be provided by multiple separate views encoded into an interlaced (composite) image, with the decoding element ensuring that one particular image is seen by the viewer depending on the viewer's viewpoint. To reduce the computational requirements for rendering images for autostereoscopic displays, a more efficient viewpoint encoding method is necessary.

- 2 -

An alternative display technique uses a conventional (monoscopic) two-dimensional display but creates a 3D impression on the viewer by updating a 2D rendering of a 3D scene based on the user's changing viewpoint, employing motion disparity as a depth cue. This occurs in dependence on a device input, where the input is representative of the viewpoint from which the viewer is observing the display. For example, in a mobile device, the viewer may tilt the device and in response the display shows a different view of the 3D entity or scene. If the update is sufficiently smooth and sensitive to the viewer's changing viewpoint, then an impression of three-dimensionality can be created. Furthermore, the user can interact with the 3D scene in an intuitive manner to inspect different views of the scene simply by tilting the device.

Typically, a source of 3D data is re-rendered in real time for each change in viewpoint to obtain the respective image to display. Real-time rendering is subject to a trade-off between complexity of the 3D scene that can be rendered, resolution of the rendered image, and responsiveness of the display system. The demand for real-time rendering of complex scenes with high resolution and good responsiveness may pose a computational challenge, particularly for mobile devices (with typically relatively low computational power). For complex scenes the display may suffer a lag between user interaction and response. This may impede intuitive recognition of the 3D nature of the scene and detract from the overall usability of the display unit.

The present invention seeks to alleviate some of the above problems.

According to an aspect of the present invention, there is provided a method of rendering images for autostereoscopic display, comprising: generating a pixel value for each pixel of an image, each pixel being mapped to a single viewpoint, such that an integer number of pixel values are generated for at least one element of a decoder array; storing each generated pixel value in a buffer matrix; resampling the pixel values stored in the buffer matrix in dependence on the number of display pixels for each element of the decoder array; and displaying the resampled pixel values. By resampling the pixel values in

- 3 -

dependence on the number of display pixels for each decoder element any misalignment between the display pixels and the decoder array can be taken into account, and therefore a higher fidelity, undistorted display may be generated.

- 5 Preferably, the integer number of pixel values is chosen to be greater than the number of display pixels for the at least one element of the decoder array. More preferably, the number of display pixels is rounded up to the nearest integer.

- 10 Preferably, the number of display pixels is rounded up to an integer number 1, 2, 3, 4, or 5 more than the number of display pixels. By increasing the number of display pixels the error associated with resampling may be reduced.

- 15 Preferably, the view point is mapped utilising ray tracing. More preferably, the pixel value is generated by integrating along the traced ray. The integration scheme utilised may be one of: emission absorption model; maximum intensity projection; and isosurface detection.

- 20 Preferably, a ray direction vector is computed directly, and in one arithmetic operation, from the view point for the each display pixel. Preferably, the ray is traced from a point a distance in front of the display to a point a distance behind the display, such that 3D object space extends in front of and behind the display surface.

- 25 Preferably, a masking operation is utilised to determine unused pixels. More preferably, the masking operation uses data volume convex hull projection. More preferably, the pixel values are not generated for the unused pixels.

- 30 Preferably, the method further comprises determining an intensity distribution from the generated pixel values, wherein the resampling utilises said intensity distribution to determine said resampled pixel values. The intensity distribution may be determined utilising linear interpolation, or a fitted polynomial curve.

Preferably, the method further comprises performing a transformation on the generated pixels before resampling. The transformation may be at least one of: linear scaling;

- 4 -

linear shift; translation; and other affine.

Preferably, the resampling is performed in dependence on at least one physical parameter of the decoder array. The at least one physical parameter may be at least
5 one of: temperature; stress; and deformation.

Preferably, the transformation is performed in dependence on at least one physical parameter of the decoder array. The at least one physical parameter may be at least one of: temperature; stress; and deformation.

10

Preferably, the resampling is performed in dependence on at least one environmental parameter. The at least one environmental parameter may be at least one of: humidity; and temperature.

15 Preferably, the transformation is performed in dependence on at least one environmental parameter. The at least one environmental parameter may be at least one of: humidity; and temperature.

Preferably, the decoder is a micro-optical array comprising a plurality of lenslet elements
20 or a parallax barrier. The plurality of lenslet elements may be substantially cylindrical or rotationally symmetrical.

Preferably, the resampling utilises a graphical processing unit.

25 Preferably, the resampled pixels are printed and/or displayed electronically.

Preferably, images are rendered for autostereoscopic display by:

computing the viewpoint position from the stored pixel position for the decoder array;
30 calculating a ray direction from the viewpoint through the stored pixel;
moving back along the ray vector from the stored pixel by a single data volume side length;

- 5 -

casting the ray through the volume for a multiple of volume side lengths, in a stepped manner, to accumulate the pixel colour value; and
adjusting the image and viewing distance parameters for display.

More preferably, the multiple is $4\sqrt{3}$.

5

Preferably, display data is subjected to affine transformation by transforming data lookup coordinates.

10 According to further aspect of the present invention, there is provided a method of image rendering, wherein a virtual composite image is constructed from a plurality of views of a three-dimensional entity, and the virtual composite image is resampled to a displayed composite image.

15 Preferably, the composite image is encoded according to a decoding element of an autostereoscopic display. The virtual composite image may be constructed in dependence of the display decoding element.

20 Preferably, the decoding element is monitored by a decoding element monitoring unit, and the virtual composite image is adapted in dependence of changes of the decoding element.

Preferably, the virtual composite image is scaled prior to resampling to display a composite image that is adapted to the scaling of the decoding element.

25 Preferably, the virtual composite image is scaled prior to resampling to display a composite image that is adapted for viewing at a position different than optical infinity.

30 Preferably, the viewing distance of a viewer is detected by a viewer detection unit, and the displayed composite image is adapted in dependence of the viewer's viewing distance.

Preferably, the virtual composite image is shifted prior to resampling to display a composite image that is adapted to a particular centre-of-parallax.

- 6 -

Preferably, the autostereoscopic display encodes a one-dimensional, or two-dimensional, distribution of viewpoints.

5 According to a further aspect of the present invention, there is provided a method of rendering autostereoscopic images, comprising:

generating a pixel value for each pixel of an image, each pixel being mapped to a respective viewpoint; and

printing the generated pixel values.

10

Preferably the view point is mapped utilising ray tracing. Preferably the pixel value is generated by integrating along the traced ray.

Examples of an integration scheme that may be utilised include: emission absorption
15 model; maximum intensity projection; and isosurface detection.

Preferably a ray direction vector is computed directly, and in one arithmetic operation, from the view point for the each display pixel.

20 Preferably the ray is traced from a point a distance in front of the display to a point a distance behind the display, such that 3D object space extends in front of and behind the display surface.

Preferably the method further comprises determining unused pixels utilising a masking
25 operation. Preferably the masking operation uses data volume convex hull projection. Preferably the pixel values are not generated for the unused pixels.

Preferably the method further comprises determining an intensity distribution from the generated pixel values, wherein the resampling utilises said intensity distribution to
30 determine said resampled pixel values. Preferably said intensity distribution is determined utilising at least one of: linear interpolation; and polynomial curve fitting.

Preferably the method further comprises performing a transformation on the generated

- 7 -

pixels before resampling. Said transformation may for example be at least one of: linear scaling; linear shift; translation; and other affine. Preferably said transformation is performed in dependence on at least one physical parameter of the decoder array.

- 5 Preferably resampling is performed in dependence on at least one physical parameter of the decoder array. Preferably the at least one physical parameter is at least one of: temperature; stress; and deformation.

10 Preferably abovementioned transformation is performed in dependence on at least one environmental parameter.

Preferably resampling is performed in dependence on at least one environmental parameter.

- 15 The at least one environmental parameter may be at least one of: humidity; and temperature.

20 Preferably the decoder is a micro-optical array comprising a plurality of lenslet elements or a parallax barrier. Preferably the plurality of lenslet elements are substantially cylindrical or rotationally symmetrical.

Preferably the resampling utilises a graphical processing unit.

25 Preferably the method further comprises: computing the viewpoint position from the stored pixel position for the decoder array; calculating a ray direction from the viewpoint through the stored pixel; moving back along the ray vector from the stored pixel by a single data volume side length; casting the ray through the volume for a multiple of volume side lengths, in a stepped manner, to accumulate the pixel colour value; and adjusting the image and viewing distance parameters for display. Preferably the multiple
30 is $4\sqrt{3}$.

Preferably display data is subjected to affine transformation by transforming data lookup coordinates.

- 8 -

According to a further aspect of the present invention, there is provided a method of image rendering, comprising: a virtual composite image composed of discrete pixels is constructed from a plurality of discrete views of a three-dimensional entity, the virtual
5 composite image is interpolated to approximate a continuous composite image that represents continuous views of the three-dimensional entity, and the continuous composite image is resampled to a display composite image that is displayed by means of an autostereoscopic display.

10 Preferably, the virtual composite image is constructed from a plurality of views of a three-dimensional entity, and the virtual composite image is mapped to a physical display, whereby the mapping scales and resamples the virtual display image to display pixels.

Preferably, the resampling accounts for a non-integer pixel-to-decoding element ratio.
15

Preferably, the scaling accounts for viewing distance for the display.

Preferably, the scaling accounts for the horizontal/vertical centre-of-parallax in the display.
20

Preferably, the scaling accounts for dynamic changes of the decoding element.

According to a further aspect of the present invention, there is provided a method of providing a display of a three-dimensional scene, comprising: receiving an input
25 representative of a given viewpoint; and, in dependence on the input, displaying an image providing a view of the three-dimensional scene corresponding to the given viewpoint, the displaying comprising: accessing data defining a viewpoint distribution representing a plurality of views of the three-dimensional scene; sampling the viewpoint distribution in dependence on the input to generate a view of the three-dimensional
30 scene corresponding to the given viewpoint; and displaying an image based on the generated view.

The term "viewpoint distribution" as used herein preferably refers to a distribution of data

values representative of a plurality (or more specifically a range) of views of the three-dimensional scene, each view corresponding to a particular viewpoint. The data defining the viewpoint distribution may comprise samples of the distribution (in which case sampling the distribution may comprise a resampling operation). The data defining the viewpoint distribution thus provides an encoding of multiple views of the three-dimensional scene, and such data is accordingly also referred to herein as a "multi-viewpoint representation". An alternative term also used herein to describe such data is "composite viewpoint representation data" since the data can represent multiple viewpoints simultaneously in a composite form. The data defining the viewpoint representation may be generated by a variety of rendering techniques, as will be discussed in more detail below.

By sampling the viewpoint distribution (for instance by resampling data samples of a multi-viewpoint representation) a display image can be generated without rendering the three-dimensional scene from scratch. This may reduce the processing time required to generate the display image and permit a quicker, more responsive display that allows real-time viewer interaction and a consistent level of usability. In particular, the processing time is dependent on the sampling operation but independent of any rendering processes required to generate the data defining the viewpoint distribution. Consequently scene rendering complexity can be decoupled from the performance of the display device, allowing great flexibility in application of industry standard image origination, processing and compositing processes in the content pre-rendering stage to produce highly realistic results.

The generated view is preferably an image, and the displayed image may simply be the generated view, or alternatively some post-processing may be performed in relation to the generated view to produce the displayed image.

Preferably, the method comprises continuously updating the displayed image in response to changes in the input representative of the given viewpoint. This continuous updating of the display can generate an impression of three-dimensionality on the viewer. As scene complexity is decoupled from the performance of the display device, complex scenes with high resolution can be displayed with good responsiveness. This

- 10 -

assists intuitive recognition of the three-dimensional nature of the scene and promotes the overall usability of the display unit.

5 Preferably, sampling the viewpoint distribution comprises convolving the viewpoint distribution with a sampling function.

10 Preferably, the sampling step comprises sampling the viewpoint distribution in dependence on one or more viewpoint parameters determined based on the input. The one or more viewpoint parameters preferably define the viewpoint corresponding to which a view is to be generated. Preferably, the one or more viewpoint parameters comprise one or more of: viewing angle information; viewing distance information; and viewpoint coordinate information. Use of an input that is representative of a viewpoint, such as viewing angle information, viewing distance information, or viewpoint coordinate information, assists intuitive recognition of the three-dimensional nature of the scene.

15 Preferably, the viewpoint distribution represents a range of continuously varying views of the three-dimensional scene. However, the distribution is preferably defined by a finite set of samples. Based on the defined distribution, it is then preferably possible (typically by resampling e.g. by interpolation) to generate an arbitrary view within the range of the defined distribution.

20 The viewpoint distribution may represent a one-dimensional range of viewpoints (e.g. corresponding to a left-to-right shift in the viewer's viewpoint). Alternatively, the viewpoint distribution may represent a two-dimensional range of viewpoints (e.g. corresponding to viewpoint shifts in both horizontal and vertical dimensions). For example, the viewpoints may be distributed on a substantially cylindrical surface surrounding (or part enclosing) the three-dimensional scene, or on some other predetermined viewpoint plane/surface. Alternatively, the viewpoint distribution may represent a three-dimensional distribution of viewpoints (e.g. combining a range of viewing distances with horizontal and vertical ranges). Alternatively, changes in viewing distance may be accommodated by scaling, as described in more detail below.

30 The viewpoint distribution preferably comprises a distribution of image data values.

- 11 -

Preferably, the data comprises data samples defining the viewpoint distribution. The data samples preferably comprise image data samples and the sampling step may then comprise resampling the image data samples.

5

The image data samples may include samples for each of a plurality of image pixels and/or for each of a plurality of pixel components (e.g. for individual colour components of the pixels). Thus, the data may in effect define a plurality of image data distributions, each distribution defining the variation of a particular pixel or pixel component in the target image (i.e. the image to be generated). The plurality of image data distributions may then together define the viewpoint distribution. The sampling operation may sample each of these distributions, typically by resampling data for each pixel or pixel component to obtain the corresponding target pixel (component) value for the view to be generated. In other words, the sampling of the viewpoint distribution may be understood as a sampling of individual distributions of pixel (component) values for an array of target pixels, with the resulting array of pixel samples forming the target view.

The image data samples preferably correspond to a plurality of pre-rendered views of the scene.

20

Thus, the data preferably comprises a plurality of pre-rendered views of the scene. For example, the views may be interlaced to form a composite, interlaced image.

Preferably, the sampling step comprises performing an interpolation based on at least two of the pre-rendered views.

Preferably, the method comprises: generating a source pixel array from the data; generating a destination pixel array based on the source pixel array, the generating step including performing a resampling operation in relation to the source pixel array; and generating an output image based on the destination pixel array.

Preferably, the sampling is performed using a texture mapping engine, preferably forming part of a graphics processing unit (GPU). This may allow fast processing using

standard tools.

Preferably, the input comprises one or more of: device orientation or motion information; viewer position or motion information; eye position information; and eye movement
5 information.

Preferably, the method is performed at a device having an image sensor, and receiving an input comprises: receiving image information from the image sensor; and analysing the image information to determine in relation to a viewer of the device one or more of:
10 viewer position information; viewer motion information; eye position information; and eye movement information. The analysis may, for example, include face or eye detection and/or face or eye tracking. Alternatively some other marker may be detected and/or tracked.

15 Preferably, the input comprises information relating to the distance between a display and a viewer, the method further comprising scaling the sampled data in dependence upon the input. The scaling may be part of the sampling function, or a separate step performed on the output of the sampling function.

20 Preferably, the method comprises generating a plurality of images corresponding to respective different viewpoints based on the viewpoint distribution, and outputting the plurality of images. The plurality of images may be displayed using a three-dimensional display system, wherein the plurality of images comprises first and second images for provision respectively to left and right eyes of a viewer. The three-dimensional display
25 system may be an autostereoscopic display. Respective separate viewpoints may be determined for the left and right eyes based on the input.

According to a further aspect of the present invention there is provided a method of generating a multi-viewpoint representation of a three-dimensional scene, comprising:
30 calculating a two-dimensional projection of the three-dimensional scene through at least one virtual encoding element onto a sampling plane; and sampling the projection in the sampling plane to generate sampled pixel data; wherein the multi-viewpoint representation comprises the sampled pixel data.

- 13 -

The term "multi-viewpoint representation" preferably refers to a data entity (e.g. data structure) which comprises data representing or defining a plurality (or range) of views of a three-dimensional scene (each view associated with a particular viewpoint), and from
5 which a plurality of such views can be reconstructed.

By projecting the three-dimensional scene through at least one virtual encoding element onto a sampling plane, a continuous viewpoint structure may be generated. With this method, no discrete, representative views of the scene are selected; instead, the
10 complete 3D scene information can be included in the multi-viewpoint representation. Therefore the parallax behaviour of the replayed scene can be smoother.

Preferably, the calculating step comprises, for each of a plurality of scene elements of the three-dimensional scene, calculating a projection of the scene element onto the
15 sampling plane through the virtual encoding element. The scene elements may for example be polygons or voxels. Preferably, the projection is calculated taking account of possible obscuration by other scene elements, for example, to ensure that only non-obscured parts of scene elements contribute to the projection. Preferably, the sampling comprises determining a contribution of the projected scene element (or non-obscured
20 part thereof) to one or more pixel samples.

Preferably, the sampling step comprises accumulating pixel sample values based on pixel sample contributions determined for each of a plurality of scene elements.

25 Preferably, the method comprises performing the calculating step for each of a plurality of virtual encoding elements. The sampling step preferably comprises accumulating pixel sample values based on pixel sample contributions determined for each of the virtual encoding elements (and preferably for each scene element projected through each encoding element). Each virtual encoding element may be associated with a
30 projection field in the sampling plane, and the projection of scene elements through a virtual encoding element may contribute only to pixel samples within its associated projection field (referred to as "clamping" in the detailed description given below).

- 14 -

Preferably, the projection is calculated based on optical characteristics of the one or more virtual encoding elements.

The term "virtual encoding element" preferably refers to a virtual representation of an encoding element (e.g. as a mathematical construct or operation, or an encoding function), which may be analogous to a physical encoding element that could be used to encode (or decode) optical information. Thus, calculating a projection through a virtual encoding element preferably involves calculating a geometric projection of a scene or scene element onto a projection plane in dependence on one or more encoding parameters or encoding characteristics (the encoding parameters or encoding characteristics being determined by the virtual encoding element). The one or more virtual encoding elements may comprise one or more virtual lenslets. Alternatively, the one or more virtual encoding elements may comprise one or more virtual parallax barriers.

15

According to a further aspect of the present invention there is provided a non-transitory computer-readable medium encoding a data structure storing a multi-viewpoint representation of a three-dimensional scene, wherein the multi-viewpoint representation comprises data defining a viewpoint distribution representing a range of views of the three-dimensional scene.

20

Encoding a data structure storing a multi-viewpoint representation of a three-dimensional scene assists the processing required to generate a viewpoint-dependent display image.

This may result in quicker processing and faster display speed.

25

Preferably, the viewpoint distribution comprises a distribution of image data values.

30

Preferably, the multi-viewpoint representation comprises image data samples defining the viewpoint distribution. The image data samples preferably correspond to a plurality of pre-rendered views of the scene.

Preferably, the multi-viewpoint representation is arranged to enable generation of arbitrary views of the three-dimensional scene within the range of views represented by

- 15 -

the viewpoint distribution.

Preferably, the multi-viewpoint representation is compressed.

5 Preferably, the data structure further comprising metadata relating to the multi-viewpoint representation. The metadata may comprise at least one of: autostereoscopic lenslet width; autostereoscopic lenslet radius of curvature; autostereoscopic lenslet thickness; number of viewing window samples; viewing window definition; and compression parameters.

10

According to a further aspect of the present invention there is provided a system for providing an image of a three-dimensional scene, comprising: means for receiving an input representative of a given viewpoint; means for, in dependence on the input, outputting an image providing a view of the three-dimensional scene corresponding to
15 the given viewpoint, the outputting means comprising: means for accessing data defining a viewpoint distribution representing a plurality of views of the three-dimensional scene; means for sampling the viewpoint distribution in dependence on the input to generate a view of the three-dimensional scene corresponding to the given viewpoint; and means for outputting an image based on the generated view.

20

Preferably, the system continuously updates the output image in response to changes in the input representative of the given viewpoint.

Preferably, the means for sampling the viewpoint distribution comprises means for
25 convolving the viewpoint distribution with a sampling function.

Preferably, the means for sampling comprise means for sampling the viewpoint distribution in dependence on one or more viewpoint parameters determined based on the input. The one or more viewpoint parameters preferably define the viewpoint
30 corresponding to which a view is to be generated. Preferably, the one or more viewpoint parameters comprise one or more of: viewing angle information; viewing distance information; and viewpoint coordinate information. Use of an input that is representative of a viewpoint, such as viewing angle information, viewing distance information, or

viewpoint coordinate information, assists intuitive recognition of the three-dimensional nature of the scene.

5 Preferably, the viewpoint distribution represents a range of continuously varying views of the three-dimensional scene. However, the distribution is preferably defined by a finite set of samples. Based on the defined distribution, it is then preferably possible (typically by resampling e.g. by interpolation) to generate an arbitrary view within the range of the defined distribution.

10 The viewpoint distribution may represent a one-dimensional range of viewpoints (e.g. corresponding to a left-to-right shift in the viewer's viewpoint). Alternatively, the viewpoint distribution may represent a two-dimensional range of viewpoints (e.g. corresponding to viewpoint shifts in both horizontal and vertical dimensions).

15 The viewpoint distribution preferably comprises a distribution of image data values.

Preferably, the data comprises data samples defining the viewpoint distribution. The data samples preferably comprise image data samples and the sampling step may then comprise resampling the image data samples.

20

The image data samples may include samples for each of a plurality of image pixels and/or for each of a plurality of pixel components (e.g. for individual colour components of the pixels).

25 Preferably, the input comprises one or more of: device orientation or motion information; viewer position or motion information; eye position information; and eye movement information.

30 According to a further aspect of the present invention there is provided a system for generating a multi-viewpoint representation of a three-dimensional scene, comprising: means for calculating a two-dimensional projection of the three-dimensional scene through at least one virtual encoding element onto a sampling plane; and means for sampling the projection in the sampling plane to generate sampled pixel data; wherein

- 17 -

the composite multi-viewpoint representation comprises the sampled pixel data.

According to a further aspect of the present invention there is provided a method of
5 providing a display of a three-dimensional scene, comprising: receiving an input
representative of a given viewpoint; in dependence on the input, displaying an image
providing a view of the three-dimensional scene corresponding to the given viewpoint,
the displaying comprising: accessing data comprising a plurality of views of the three-
dimensional scene; convolving the data with a sampling function to generate a view
10 image from the given viewpoint; and displaying an image based on the generated view
image.

According to a further aspect of the present invention there is provided a method of
providing a display of a three-dimensional scene, comprising: receiving an input
15 representative of a given viewpoint; in dependence on the input, displaying an image
providing a view of the three-dimensional scene corresponding to the given viewpoint,
the displaying comprising: accessing image data comprising a plurality of views of the
three-dimensional scene; interpolating between at least two of the plurality of views to
derive an interpolated view; and displaying an image based on the interpolated view.

20

The invention extends to methods and/or apparatus substantially as herein described
with reference to the accompanying drawings.

The invention also provides a computer program and a computer program product for
25 carrying out any of the methods described herein and/or for embodying any of the
apparatus features described herein, and a computer readable medium having stored
thereon a program for carrying out any of the methods described herein and/or for
embodying any of the apparatus features described herein.

30 The invention also provides a signal embodying a computer program for carrying out any
of the methods described herein and/or for embodying any of the apparatus features
described herein, a method of transmitting such a signal, and a computer product having
an operating system which supports a computer program for carrying out any of the

- 18 -

methods described herein and/or for embodying any of the apparatus features described herein.

Any apparatus feature as described herein may also be provided as a method feature,
5 and vice versa. As used herein, means plus function features may be expressed alternatively in terms of their corresponding structure, such as a suitably programmed processor and associated memory.

Any feature in one aspect of the invention may be applied to other aspects of the
10 invention, in any appropriate combination. In particular, method aspects may be applied to apparatus aspects, and vice versa. Furthermore, any, some and/or all features in one aspect can be applied to any, some and/or all features in any other aspect, in any appropriate combination.

15 It should also be appreciated that particular combinations of the various features described and defined in any aspects of the invention can be implemented and/or supplied and/or used independently.

Furthermore, features implemented in hardware may generally be implemented in
20 software, and vice versa. Any reference to software and hardware features herein should be construed accordingly.

These and other aspects of the present invention will become apparent from the
following exemplary embodiments that are described with reference to the following
25 figures in which:

Figure 1 shows the basic principles that underlie autostereoscopic displays;

Figure 2 shows a pixel array with a lenticular lens decoder array;

Figure 3 shows a lenticular lens decoder array with cylindrical lenses;

30 Figure 4 shows a lenticular lens decoder array with rotationally symmetrical lenses;

Figure 5 shows volume ray-casting;

Figure 6 shows a realistic display with irregular pixel-viewpoint association;

Figure 7 shows resampling of a virtual composite image to a displayed composite image;

- 19 -

- Figure 8 shows a linear scaling transformation applied to the virtual composite image prior to resampling it to a displayed composite image for accommodation of proximity viewpoints, for an ideal display;
- Figure 9 shows a linear shifting transformation applied to the virtual composite image prior to resampling it to a displayed composite image for accommodation of centre-of-parallax, for an ideal display;
- 5 Figure 10 shows implementation of a volume ray-casting algorithm;
- Figure 11 shows a flow diagram of the ray-casting process;
- Figure 12 shows the use of graphics tools to sample composite viewpoint representation data and generate a display image;
- 10 Figure 13 shows a different particular selection of pixels that relate to a different viewpoint being drawn onto a destination pixel array;
- Figure 14 shows an example where a new image is generated as a combination of source pixel values;
- 15 Figure 15 shows a source quad being sampled onto a destination pixel array;
- Figure 16 shows the relationship between the viewer angle and the mapping of the source quad and destination pixel array relative to one another;
- Figure 17 shows the core scheme of forward mapping;
- Figure 18 shows the core scheme of forward mapping;
- 20 Figure 19 shows tilt compensation in the case of a static parallax barrier;
- Figure 20 shows tilt compensation in the case of a lenslet array;
- Figure 21 shows a file format that facilitates generating a display image;
- Figure 22 shows a process diagram for generating a display image;
- Figure 23 shows a typical graphics pipeline; and
- 25 Figure 24 shows components of a device utilising the method of generating an image.

Rendering Images for Autostereoscopic Display

- 30 Figure 1 shows the basic principles that underlie autostereoscopic displays. A rectangular array of independently addressable colour square (generally rectangular) pixels 100 is overlaid by a decoding element. The decoding element is a micro-optical array that enables the viewer 104 to see only certain pixels from a given viewing position

- 20 -

106. In the case of a lenticular lens decoder array 102 an array of small lenses 108 (lenslets) is formed in a substrate, typically a polymer. The pixel array 100 is coincident with the rear focal distance 110 of the lenses 108, so that, in the ideal case, a focussed spot at the pixel plane 114 projects a parallel beam 200 at a given angle 202. This is shown in more detail in Figure 2. The array of lenses 102 serves to decode the information placed at the pixel plane 114. From a defined viewpoint 106-1, a defined set of pixels 112-1 is seen by the viewer 104. If the viewer 104 moves to a different viewpoint 106-2, a different image composed of a different subset of pixels 112-2 is seen by the viewer 104. This can be employed to generate a volumetric optical model that may be viewed and perceived as a 3D image with full, natural parallax (look-around). The image at the display is thus a composite image that represents a plurality of views of a scene from multiple viewpoints.

The lenses may be cylindrical, as in Figure 3, in which case lens action, hence image decoding and parallax, take place only in one dimension. Figure 3 shows a horizontal-only parallax decoder array with vertical cylinder alignment. Alternatively the lenses may be rotationally symmetrical, as in Figure 4, giving omnidirectional lens action and viewable parallax in all directions.

Instead of the lenticular lens decoder array described above, alternative types of decoder arrays, such as parallax barrier arrays, as are well known in the art, can be used to generate autostereoscopic displays.

To render a wide range of volumetric data sets, volume ray-casting (VRC) is a popular technique. Examples of volumetric data sets include CGI/CAD models and scenes, multiple 2D slice-based (tomograms) or 2D+DepthMap data sets, 3D/4D ultrasound data, and 3D TV/video signals. VRC is an image-orientated scheme that makes use of pixel-parallel computing capabilities of modern off-the-shelf Graphics Processor Units (GPUs). A suitable method of combining 3D ultrasound and electrical impedance measurements for autostereoscopic display is disclosed in GB patent application no. 1106110.8 filed 11 April 2011 and titled "Bio-medical visualisation", the contents of which are incorporated herein by reference.

- 21 -

Volume ray-casting is illustrated in Figure 5. The system casts rays 502 from a (virtual) viewpoint 106 in front of the display 100, through each display surface pixel 504, and then through to the (virtual) location of a volumetric data set 500. The ray 502 is stepped through the data set 500, integrating contributions to the colour and intensity of the pixel 504 at each step location, until it has exited the far side of the volumetric data set. Many integration schemes can be used to control the final pixel colour, for example the emission/absorption model which takes into account voxel opacity to enable translucent rendering, the maximum intensity projection as commonly used in X ray applications, or isosurface detection. Other integration schemes are possible too.

5

As illustrated in Figure 1, in the autostereoscopic displays described there is a clear relationship between viewpoint and the pixels associated with the viewpoint. In applying the VRC technique, each pixel has a viewpoint associated with it. In Figures 1 and 2 an idealised situation is illustrated, whereby the lenslet-pixel ratio is an integer, and each lenslet edge aligns perfectly with a pixel edge. A more realistic situation is illustrated in Figure 6. In Figure 6 the pixel-viewpoint association is more complex than for the idealised situation 610. Pixels are not always in the same relative positions within their respective lenslet. Neither VRC nor any other viewpoint-dependent rendering techniques can be applied directly.

10

The calculated VRC intensity distribution across the pixels of a display is only a discrete sample of a full-information integral image that contains an angular continuum of viewpoints. Therefore, it is possible to calculate the intensity distribution for a virtual, idealised lenslet-pixel association, and then resample the obtained intensity distribution to obtain an intensity distribution suitable for the real, non-ideal lenslet-pixel association 720. This concept is illustrated in Figure 7. First, a virtual, idealised lenslet-pixel association 710 is assumed. The real decoding array lenslet configuration is known (from a product datasheet, for instance, or from measurement) and is chosen for the virtual, idealised lenslet-pixel association 710. The actual, non-integer number of pixels appearing behind each lenslet on the decoding array is rounded up to the nearest integer, producing a viewpoint count. Alternatively, the non-integer number of pixels is rounded up to an integer number 1, 2, 3, 4, 5 or more, more than the non-integer number of display pixels behind each lenslet element of the decoder array.

15

20

25

30

- 22 -

In Figure 7 for example, the actual lenslet-pixel ratio is 1:4.3, and the viewpoint count is 5. The VRC procedure is applied to generate a virtual composite image for the virtual, idealised lenslet-pixel association 710, with an intensity value 700 for each virtual pixel 5 702. Next, the intensity distribution 704 is taken as a sample of the continuum intensity distribution 706 for infinite viewpoints and infinite pixels. The continuum intensity distribution 706 can be approximated by linear interpolation of the discrete intensity values 700 of the virtual composite image, or it can be approximated by other models such as a fitted curve. Finally, the intensity distribution 714 for the real pixels 712 is 10 found from the continuum intensity distribution 706. By increasing the number of pixel values generated for each lenslet element a smoother intensity distribution (pixel value distribution) can be generated, and so the error associated with the linear interpolation between data points can be reduced, but with increased computation time. Although fewer virtual than display pixels cause problems, more virtual pixels do not improve final 15 image quality. If more display pixels are available, quality is improved by using more render pixels (and hence more viewpoint samples).

In one example the volume ray-casting (VRC) is implemented using the pixel-parallel computing capabilities of a Graphics Processor Unit (GPU). This permits operation in 20 real-time without requiring specialist graphics hardware. The real-time version executes entirely in GPU hardware. If real-time display is not required, for instance for printing applications, a CPU-only or a CPU/GPU hybrid implementation may be used. By using the GPU's texture manipulation facilities, the virtual image is mapped internally to a rectangular primitive (a 'quad') which is drawn to display memory. The mapped quad is 25 then scaled in real time using the GPU's geometric transformation facilities. This results in a filtered resampling using built-in interpolation facilities of the GPU's texture engine. The display image is now correctly aligned with the autostereoscopic display decoder lenslets. Because it uses techniques for the execution of which the GPU is highly optimised, this results in a miniscule and negligible performance penalty.

30

The image rendering procedure as described is equally applicable to parallax barrier arrays as to the lenticular lens decoder array. In the case of a parallax barrier array, barrier slits replace the decoder lenslets, with all the other features regarding alignment

- 23 -

and resampling remaining same. Other decoding elements may perform the function of associating different pixels with different viewpoints.

5 A lenticular decoding array or other uni-directionally selective means can be used with a horizontal group of viewpoints to provide a reduced information integral image with horizontal parallax only. Alternatively a square- or hexagonally-based full parallax microlens array can be used as a decoder. With a two dimensional matrix of origin viewpoints this provides full omni-directional parallax in the display, so that all-around (rather than just horizontal) look-around is possible.

10

For higher resolution viewpoint sampling when targeting a display surface comprised of spatially distinct colour component sub-pixels, rays may be cast separately from each colour component sub-pixel. Additionally, multiple rays can be cast from each pixel and integrated, for a still greater viewpoint resolution gain, resulting in smoother look-around
15 (effectively a form of oversampling).

Figure 8 illustrates how, in the description so far, the viewpoints have been assumed to be at a great distance from the display, at optical infinity 106-1, defined in that the viewing direction from a viewpoint is considered parallel for all pixels of the display. If
20 however the viewpoints 106-2 are closer to the display, the pixel group relating to a lenslet must be slightly wider than the lenslet itself for the viewing rays to converge correctly. This can be accommodated with ease in the procedure of resampling the virtual composite image to a displayed composite image. A linear scaling transformation
802 is applied to the virtual composite image prior to resampling it to a displayed
25 composite image. The linear scaling of the virtual intensity distribution allows the displayed composite image to be slightly wider than for infinity viewpoints. Figure 8 illustrates an ideal lenslet-pixel alignment with display pixels perfectly aligned with decoder elements, analogous to the idealised situation 610 in Figure 6. The ideal lenslet-pixel alignment is for illustration purposes only, and it is obvious that the concept
30 of scaling the virtual intensity distribution for non-infinity viewpoints is equally applicable to a realistic lenslet-pixel alignment with a non-integer number of pixels appearing behind each lenslet. In this case a virtual idealised lenslet-pixel association is used to

- 24 -

generate the virtual intensity distribution which is scaled before being resampled to the real lenslet-pixel alignment.

5 The above procedure of correcting the displayed composite image for a particular viewing distance can be adapted in response to a viewer positioning himself in front of the display. Detection of an actual viewer position can be implemented for instance by a webcam, as is frequently built into displays. In this manner the optimum viewing distance of the display is adapted in dependence of an actual viewer position, in real-time.

10

Under some circumstances the configuration of display and decoder material may shift. For instance thermal expansion, physical stress, any external forces applied to the decoder could cause the decoder material to expand and shift. Alternatively, or in addition, environmental parameters, such as humidity and temperature, can also cause the decoder material to expand and shift; for example, the decoder array may expand/contract due to changing humidity. This can be accommodated in the procedure of resampling the virtual composite image to a displayed composite image. A linear scaling transformation is applied to the virtual composite image prior to resampling it to a displayed composite image. In this manner, automatic compensation for shifts of the configuration of display and decoder material may be implemented. For instance in the case of thermal decoder material expansion, the temperature of the display surface is measured via a built-in sensor, and the display pixel matrix scale factor adjusted to compensate for decoder array expansion using known material characteristics.

25 Figure 9 illustrates how manipulation of the horizontal/vertical centre-of-parallax (centre of angular viewing zone) in the display can be accommodated in the procedure of resampling the virtual composite image to a displayed composite image. The centre-of-parallax is normally in the physical centre line of the display 854, but there are advantages in installation flexibility in certain applications by being able to re-centre the display 556. This then gives the viewer the impression that he is looking at the central view of the scene, even if he is actually not looking at the display from a central position. For example, the centre-of-parallax may be adapted if a fixed obstacle is positioned in the normal viewing zone of the display. Alternatively, if the viewer has both a 2D and 3D

- 25 -

monitor, then the 3D monitor may be positioned for a fixed off-centre viewing. As seen in Figure 9, adjustment of the centre-of-parallax can be obtained by applying a linear shift transformation to the virtual composite image prior to resampling it to a displayed composite image.

5

Implementation of the volume ray-casting either on a GPU (with parallel pixel processing) or a CPU follows the following basic algorithm per pixel, as illustrated in Figure 10:

- 10 1. Compute the viewpoint position 106 from the current (virtual) pixel 908 for the specific applied decoding element 910;
2. Calculate the ray direction 906 from the viewpoint 106 through the current pixel 908;
3. Move back along the ray vector from the display pixel 908 by one data volume side length to ensure the start point 904 is always at a point in front of the data volume 900;
- 15 4. Cast ray through the volume for $4\sqrt{3}$ volume side lengths (to ensure it terminates behind the data volume if rotated), in a stepped manner, accumulating the pixel colour value;
5. Adjust image and viewing distance parameters for display and application.

20

Figure 10 illustrates that, in general, the ray start point 904 can be in front of the virtual encoder array/render pixel surface arrangement 710, resulting in content appearing in front of the 3D display unit screen plane. In the illustration approximately 1/3 of the data volume (the object space) is in front of the display, and 2/3 is behind the display.

25 Standard ray casting schemes assume that the object space is completely behind the display screen plane (there is no in-front/behind concept in a 2D display).

30

The algorithm described above is executed for every pixel on the render pixel surface, although some of these may not actually have any contribution whatsoever from the data volume since the associated viewpoint rays completely miss it. In these cases the full stepped ray cast will still be performed, wasting computational resources and thus reducing rendering performance. To prevent this, the system may create a mask from the data volume content boundaries, which can be used to determine whether or not a

- 26 -

render surface pixel should be processed (and hence result in a ray cast), thus improving efficiency by avoiding redundant ray trace processing completely.

The method consists of the following steps:

- 5 1. Internal projection of all corner points of the content boundary of the data volume to the render pixel surface from both extremes of the angular viewing area;
2. The computation of the convex hull of the resulting points to produce the boundary of the area including all data volume content when viewed from any position within the angular viewing area;
- 10 3. Drawing of a filled polygon with the boundary determined above to an internal memory surface, to create a processing mask which is known as the operation region;
4. The application of the processing mask within the graphics framework in use so that only masked render surface pixels are processed. There are general facilities
- 15 which can be used for this in standard graphical programming frameworks (e.g. OpenGL). An example is the stencil buffer, which can contain a value corresponding to each pixel in the final render surface. Values can be written to the stencil buffer and then tests set up to render/not render a pixel depending on the value encountered.

20

The generation of pixel values for a true 3D display using a lenslet decoder array requires that a complementary lenslet encoder array is modelled in the rendering process. Rays are traced from positions on a continuum of possible viewpoints, through the modelled encoder array and a memory-based (virtual) render pixel surface. While

25 being traced they impinge on a data volume that contains the three dimensional information distribution to be rendered.

It is a requirement of the system that the final displayed, classified representation of the data volume may be subject to affine transformations (rotation, translation, scaling) to

30 allow the viewer to manipulate the view. It is normal in 3D computer graphics for this to be achieved by transformation of vertex coordinates within the world space. However, in the system the virtual encoder array and viewpoints must remain static and

- 27 -

untransformed as they exist in viewer space. Normally in 3D computer graphics, object and viewer space are equivalent but in opposite sense of one another.

It is therefore necessary to be able to separate object and viewer spaces. This is
 5 achieved by defining the virtual encoder lenslet array and viewpoint array in viewer world space as normal, but transforming the data volume lookup coordinates according to the transformation required by the viewer. The lookup coordinates are transformed in an inverse sense.

10 In the following, the ray casting process is described in more detail. The individual steps are summarised in Figure 11. In the description, bold type is used to denote vector quantities. Vector quantities may be used to hold spatial coordinates (x, y, z, w) or colours (r, g, b, a - red, green, blue and opacity components).

15 To render a surface pixel, the first step 1102 is to compute the viewpoint origin position 106 (in fixed viewer space) according to the (virtual) render pixel number. The current operation pixel 908 from the render pixel surface is stored in **RenderPixelIndices**. The number of pixels mapping to one lenslet in the modelled encoder array is stored as **RenderPixelsPerLenslet_{Horiz}** (horizontally) by **RenderPixelsPerLenslet_{Vert}** (vertically, only
 20 applicable if an omnidirectional parallax display is used, with for instance a square-based microlens array or two perpendicular cylindrical lens arrays). The viewpoint origin position is computed that corresponds to the exact centre of the render pixel. For horizontal-only parallax (e.g. using a cylindrical lenslet decoder) this is:

ViewpointPos. x = RenderPixelIndices. x mod RenderPixelsPerLenslet_{Horiz}
 25 **ViewpointPos. y = 0**
ViewpointPos. z = 0

For omnidirectional parallax (e.g. using a square-based symmetrical microlens decoder):

ViewpointPos. x = RenderPixelIndices. x mod RenderPixelsPerLenslet_{Horiz}
 30 **ViewpointPos. y = RenderPixelIndices. y mod RenderPixelsPerLenslet_{Vert}**
ViewpointPos. z = 0

- 28 -

The next step 1104 in the ray casting process is computing the ray direction vector 902 (in fixed viewer space) according to the viewpoint origin 106, and then computing the transformed stepped direction vector for tracing. The screen plane ray position vector, **RayScreenPos**, is proportional to the current render pixel index across the screen. This is the position, in fixed viewer space, at which the viewpoint ray crosses the render pixel surface. With the **RayScreenPos** vector the normalised ray direction vector (in viewer space) is computed:

$$\mathbf{RayDirection} = \mathbf{RayScreenPos} - \mathbf{ViewpointPos}$$

10 In preparation for efficient ray tracing, a stepped direction vector is generated to enable ray advancement through the data volume of a certain **StepSize**. This is also subjected to any necessary view transformations via matrix $T_{V \rightarrow D}$ to place it in data volume space:

$$\mathbf{RayDirDelta}_{DataSpace} = T_{V \rightarrow D} \cdot \mathbf{RayDirection} \cdot \mathbf{StepSize} .$$

15 The next step 1106 is to compute the ray start point 904 (in fixed viewer space) in front of data volume 900. As described above, the ray start point can be in front the virtual encoder array/render pixel surface arrangement. The ray start point 904 is computed so that it is guaranteed to be in front of the (possibly transformed) data volume 900, by travelling back from **RayScreenPos** by the correct proportion of the total ray cast length according to the depth of the data volume that is to appear in front of the plane of the final 3D display. In the following, **ProplnFront** is the proportion of the transformed data volume displayed in front of the screen plane (commonly 0.5 for half in front and half behind, or 0.33 for 1/3 in front and 2/3 behind); **RayCastLen** is the total length of the ray cast through the data volume, and is typically $4\sqrt{3}$ for a cubic, Cartesian data volume.

20

25 Then, the ray start point in viewer space is:

$$\mathbf{RayStartPoint} = \mathbf{RayScreenPos} - (\mathbf{ProplnFront} \cdot \mathbf{RayCastLen}) \cdot \mathbf{RayDirection} .$$

For efficiency, this is transformed into data volume space as for the stepped direction vector:

30
$$\mathbf{RayStartPoint}_{DataSpace} = T_{V \rightarrow D} \cdot \mathbf{RayStartPoint} .$$

The data volume lookup coordinates for the first pass of the inner tracing loop 1108 are the ray start point:

- 29 -

RayLookupPos_{DataSpace} = RayStartPoint .

Once these operations have been performed for the current operation pixel 908, a
 5 nested loop procedure commences to trace the ray to the rear of the data volume in
 stepped manner. The following steps are looped until the full ray cast length has been
 covered. The initial value of the accumulated final render pixel colour is FinalColour.rgba
 = (0,0,0,0).

10 The first step 1110 in the ray-tracing sub-loop 1108 is to look up data volume value(s)
 using transformed lookup coordinates:

DataValues = 3d_texture_lookup(RayLookupPos_{DataSpace})

For example if two different data sets are combined to provide e.g. opacity values and
 15 colour values, then:

OpacityValue = Opacity_data_texture_lookup(RayLookupPos_{DataSpace})

ColourValue = Colour_data_texture_lookup(RayLookupPos_{DataSpace}).

If the values obtained from the look up step are not directly opacity and colour values,
 20 then a value classification step 1112 may be performed. This step may also be a
 combined classification operation to determine the contribution to current render pixel.
 Each of the data values contained in the current value vector can contribute both colour
 and opacity (alpha) components to the current render pixel colour and opacity. To
 compute this, colour (or opacity) contributions are looked up for the data values in a
 25 colour (respectively opacity) table and assigned to the destination vector.

**ClassifiedColour.rgb = Datatype1_colour_map_lookup(Datatype1Value) +
 Datatype2_colour_map_lookup(Datatype2Value)**

**ClassifiedColour.a = Datatype1_opacity_map_lookup(Datatype1Value) +
 Datatype2_opacity_map_lookup(Datatype2Value)**

30

Soft thresholding may be applied to the opacity component. In soft thresholding, a
 transfer function is used with a linear rather than step transition. It is parameterised by

- 30 -

the data values at the lower and upper limits of the linear transition region. An example implementation is the GLSL 'smoothstep()' function.

ClassifiedColour may be determined in many ways according to the needs of the application and type of data being rendered, using techniques known in the computer
5 graphics field.

The next step 1114 in the ray-tracing sub-loop 1108 is to accumulate the current render pixel colour. The emissive classified colour r,g,b components determined above are premultiplied by the associated opacity value a, and the result used to contribute to the
10 final accumulated render pixel colour:

$$\mathbf{FinalColour.rgb} = \mathbf{FinalColour.rgb} + (1 - \mathbf{FinalColour.a}) \cdot \mathbf{ClassifiedColour.rgb} \\ \cdot \mathbf{ClassifiedColour.a}$$

$$\mathbf{FinalColour.a} = \mathbf{FinalColour.a} + (1 - \mathbf{FinalColour.a}) \cdot \mathbf{ClassifiedColour.a}$$

15 The next step 1118 in the ray-tracing sub-loop 1108 is to step to next data lookup position. The lookup position in transformed data space is advanced by adding the partial, transformed (stepped) direction vector to the current position:

$$\mathbf{RayLookupPos}_{\text{DataSpace}} = \mathbf{RayLookupPos}_{\text{DataSpace}} + \mathbf{RayDirDelta}_{\text{DataSpace}}$$

20 The ray-tracing sub-loop 1108 is now repeated by returning to first step 1110 looking up data volume value(s) for the updated lookup position.

Viewpoint-Dependent Display of Three-Dimensional Scenes

25 The above description focuses on the rendering of images for display on an autostereoscopic display. An alternative approach will now be described in which a 3D effect is simulated on a 2D display. The various concepts and techniques (e.g. rendering techniques) discussed above may be adapted as appropriate for use with this alternative approach.

30

Embodiments of the present invention provide a display technique for 3D data (entities or scenes), in which a different image is shown to the viewer in dependence on the

- 31 -

viewer's viewpoint, as determined by some input (e.g. a tilt sensor or camera tracking). In one embodiment, 2D images of the 3D scene are generated and displayed on a conventional 2D-display. The display is continuously updated as the viewer's viewpoint changes to present a different view of the 3D scene. The continuous updating of the scene produces an impression of three-dimensionality on the user, and the user can inspect different aspects of the scene by changing his viewpoint.

Rather than rendering each viewpoint image in real-time, the present approach is based on a pre-generated multi-viewpoint representation of the 3D scene. The multi-viewpoint representation encodes multiple views (or viewpoint images) of the 3D scene corresponding to a defined range of viewpoints. Once generated, the multi-viewpoint representation may be "replayed" to produce arbitrary views of the scene within the encoded range of views without the need to render the views from scratch.

More specifically, the range of viewpoint images may be encoded in area sampled composite viewpoint representation data, where the samples are representative of a continuum of viewpoints. The composite viewpoint representation data thus encodes a continuum of viewpoint images. This continuum of viewpoint images is also referred to herein as a viewpoint distribution. The viewpoint distribution may be made up of distributions of pixel values for a plurality of target replayed angular image segments which are then interrogated by the viewer. In some embodiments (e.g. when interlacing a pre-rendered set of viewpoints or ray casting/tracing from distribution pixels out into scene) the viewpoint distribution may be made up of distributions of pixel values for a plurality of target image pixels. The composite viewpoint representation data may then comprise pixel value samples which define those pixel value distributions.

A particular viewpoint image may be generated from the composite viewpoint representation data in dependence upon a viewer's viewpoint and displayed to the viewer by sampling the viewpoint distribution.

30

The sampling of the viewpoint distribution involves resampling (the composite viewpoint representation data with a sampling function. The sampling function can also be understood as a replay function which is used to replay the composite viewpoint

- 32 -

representation data to generate a particular view (where the replay function defines the sampling characteristics used to sample the data). The sampling function is analogous to the optical function of a decoding lenslet array in a lenticular display. The sampling function may be defined in relation to the characteristics of a decoding lenslet array and
5 the angle of the desired viewpoint.

The sampling of the viewpoint distribution is unlike the technique of morphing. Morphing requires warping of a first image into a second image, usually by means of a mesh with defined control points. The sampling of the viewpoint distribution requires no warping,
10 and is consequently of lower computational complexity than morphing.

It should be noted that terms such as "multi-viewpoint representation", "composite viewpoint representation data", "data defining a viewpoint distribution" and the like are used interchangeably herein unless otherwise required by context.
15

The process required to generate a display image from pre-rendered composite viewpoint representation data by resampling is far less computationally demanding than continuously re-rendering an entire scene in real-time as the viewpoint changes, and therefore allows a more responsive display to be provided without the need for advanced
20 rendering capabilities in the device, whilst still providing high image quality. This in turn allows provision of a more convincing and consistent 3D display to the viewer and hence enhanced usability of the 3D display device. Importantly, the approach decouples scene complexity completely from display performance, and therefore enables a wider range of 3D media origination processes which can provide richer and more realistic displays
25 than are possible using real time computer graphic techniques.

It should be noted that image data may be represented using a wide variety of encoding schemes. Monochrome image data may simply include intensity/brightness values for each pixel. Colour image data typically includes multiple colour components per pixel,
30 which may be encoded using a variety of colour encoding schemes / colour spaces. For example, colour may be encoded using HSL (hue, saturation, lightness) colour components or RGB (red, green, blue) colour components. It should therefore be understood that any reference herein to pixel values, colour values, intensity values, or

- 33 -

the like may in fact refer to any suitable type of image / pixel data using any appropriate encoding. Similarly where reference is made to intensity distributions, distributions based on other types of image / pixel data may be substituted.

5 A particularly useful and efficient method of implementing the above-mentioned display uses the texture engine of standard graphics tools (e.g. hardware, for instance a Graphics Processing Unit (GPU)) to perform appropriate sampling of the stored composite viewpoint representation data to generate a display image. This is illustrated in Figure 12. The composite viewpoint representation data 1201 is composed of pixels,
10 each of which is mapped to a respective range of possible viewpoints. In the example illustrated here, each pixel centre maps to one of 5 views; and the composite viewpoint representation data 1201 contains 5 complete view ranges 1202, 1204, 1206, 1208, 1210 of the 3D scene. As the pixels of the composite viewpoint representation data 1201 are merely area samples of the continuous viewpoint distribution, there is no need
15 for them to map to regularly spaced viewpoints or indeed to the same viewpoints.

The composite viewpoint representation data 1201 is drawn internally onto a rectangular primitive (a 'quad') 1200. This quad is an internal image representation, and is used for internal image processing. In Figure 12 only one pixel row of the quad 1200
20 corresponding to pixel row 1203 of the composite viewpoint representation data 1201 is illustrated, for simplicity. The source quad 1200 is mapped (using the rendering engine) onto a destination pixel array 1212. The destination pixel array 1212 is used as the display image 1218 (though post-processing may also be performed).

25 In the simplified example illustrated in Figure 12, each of the 5 complete views 1202, 1204, 1206, 1208, 1210 has 3 pixels per row. For 5 views the composite viewpoint representation data 1201 has 15 pixels per row 1203, the source quad 1200 has a mapped pixel array with 15 pixels per row, and the destination pixel array 1212 has 3 pixels per row. The display image 1218 has 3 pixels per row 1207. The pixel values of
30 the destination pixel array are generated in dependence on the values of the pixel array mapped to the source quad.

To draw the source quad onto the destination pixel array the texture engine samples

- 34 -

from the pixel array mapped to the source quad onto the destination pixel array. In Figure 12, from the source quad 1200 (that is: the composite viewpoint representation data mapped to the source quad 1200) a particular group of pixels 1206 is selected and drawn onto the destination pixel array 1212. The group of pixels 1206 relate to the same
5 viewpoint. In Figure 13, a different particular selection of pixels 1204 (relating to a different viewpoint) is drawn onto the destination pixel array 1212.

Figures 12 and 13 show how the source quad 1200 relates to the lenticular lens decoder array 102 of a lenticular autostereoscopic display. The mapping from the source quad to the destination pixel array 1212 is analogous to the viewer position angle 1220 in an
10 autostereoscopic display. The destination pixel array has the same number of pixels (in the view decoding direction) as there are lenslets represented in the original composite viewpoint representation data. In the autostereoscopic display, the lenticular array determines the final image (selected from the source composite viewpoint representation data) that is seen by the viewer. In the method described here, the mapping from the
15 source quad to the destination pixel array 1212 determines the final image seen by the viewer. In this manner the described method can effectively simulate an autostereoscopic display.

Figure 14 shows an example where none of the available interlaced images 1202, 1204,
20 1206, 1208, 1210 is drawn onto the destination pixel array 1212. Instead, a new image is generated as a combination of source pixel values. This is possible because the intensity/colour values of the source quad (and the composite viewpoint representation data) can be understood as samples of the continuum intensity distribution 706 for infinite viewpoints and therefore infinite interlaced images. The continuum intensity
25 distribution 706 can be approximated by linear interpolation of the discrete intensity values 700 of the composite viewpoint representation data 1201, or it can be approximated by other models such as a fitted curve. This linear interpolation or fitted curve may also be generated with a sampling function that is associated with the simulation of an autostereoscopic display.

30

The texture engine is configured to sample and interpolate (resample) from the source quad 1200 onto the destination pixel array 1212 with a pre-determined sampling function 1214. The source is selected only within a predetermined sampling width 1216, and

- 35 -

according to a predetermined function. Advantageously the sampling width 1216 is equal to the width of the (lenslet) segments 1222 of the composite viewpoint representation data. The sampling function can be a rectangular function 1214 for linear interpolation. The function can take other shapes to effect non-linear interpolation, for example a triangular function 1218. This might be required for different effective replay lenslet simulations. In the example illustrated in Figure 14, the intensity values of the two images 1206 and 1208 are combined, each (in this example) with respective proportions of approximately 70% respectively 30% for the rectangular function; or approximately 82% respectively 18% for the triangular function.

10

The continuum intensity distribution 706 (also referred to as a viewpoint distribution) is a distribution of data values representative of a plurality (or more specifically a range) of views of the three-dimensional scene, each view corresponding to a particular viewpoint.

The composite viewpoint representation data 1201 is used to represent the continuum intensity distribution 706. The data defining the viewpoint distribution (the composite viewpoint representation data 1201) may comprise samples of the distribution, such as discrete intensity values 700. In this case sampling the distribution may comprise a resampling operation.

20 As shown in Figure 15, the source quad 1200 is sampled and interpolated (resampled) onto the destination pixel array 1212 only within the sampling width 1216. Therefore how the source quad 1200 is mapped to the destination pixel array 1212 determines which part of the source quad 1200 is sampled onto the destination pixel array 1212. Because of the arrangement of the images from different views in the source quad, the mapping from the source quad to the destination pixel array determines which image corresponding to a particular view is generated on the destination pixel array. The upper and the lower row in Figure 15 each show a respective configuration for generating different display images. In other words, what in a lenticular (or parallax barrier) autostereoscopic display would be the viewing angle of the viewer can be used as an input to manipulate how the source quad 1200 is mapped to the destination pixel array 1212, and in doing so determine which image corresponding to a particular view is generated.

25

30

- 36 -

The mapping can be modified in dependence upon a tilt orientation or other input to achieve a change in viewpoint.

The texture engine is optimised for drawing a source quad onto a destination pixel array (and performing the necessary sampling and interpolation) and can do so very efficiently.

5 In this manner the pre-rendered composite viewpoint representation data can be processed very quickly to generate a display image. This allows real-time display of images as the viewer angle changes. Because new images are generated from the pre-rendered image by interpolation as a simulation of the optical action of lenslets in an auto-stereoscopic display, the display does not jump from one image to the next, but
10 shows smooth transitions as does the equivalent autostereoscopic display. This allows very natural and convincing representation of a 3D scene.

It is noteworthy that, unlike other 3D display methods, the display method described here does not render the image from the 3D data scene at the time of viewing, but only
15 generates a display image from the (pre-rendered) composite viewpoint representation data. This may reduce the processing time required to generate the display image and permits a quicker, more responsive display that allows real-time viewer interaction and a consistent level of usability. In particular, the processing time can be essentially constant, being determined only by the complexity of the sampling operation, and is
20 independent of the way the scene has been produced or the complexity of the scene. Consequently scene complexity is decoupled from the performance of the display, allowing great flexibility in application of industry standard image origination, processing and compositing processes in the content pre-rendering stage to produce highly realistic results.

25

In reference to a lenticular autostereoscopic display, the method of determining the position or mapping of the source quad and the destination pixel array relative to one another relates to the viewer angle as follows and as illustrated in Figure 16, where:

30 x_L is the shift distance away from the lenslet midpoint 1600 (respectively the shift distance of the source and destination quad relative to one another);

α is the angle between the viewing direction and the orthogonal to the viewing surface 1602;

R is the radius of curvature of the lenslet 1604;

- 37 -

t is the lenslet thickness 1606 at the lenslet midpoint; and
 P_L is the width of the lenslet 1608.

Similar to an autostereoscopic display, the composite viewpoint representation data only
5 represents a pre-determined range of viewpoints. The multiple data associated with
each (virtual) lenticule relate to the range of viewpoints. At the edge of the viewpoint
range a discontinuity occurs, inasmuch as neighbouring segments within the composite
viewpoint representation data relate to viewpoints that are at either extreme of the
viewpoint range, and are not necessarily closely related to one another. In the case of
10 the viewpoint-dependent display, as the mapping of the sampling function approaches
the end of the viewing range, there may be a provision to prevent the sampling function
from moving on into the next lenticule.

For example, if the viewing angle input exceeds a certain value (that corresponds to the
15 end of the viewing range), then the sampling function may be redirected to the other
extreme segment in the same (virtual) lenticule. In this manner the display flips from the
end of the viewing range back to the beginning of the viewing range. Alternatively, if the
viewing angle input exceeds a maximum, then the sampling function may be fixed at the
last segment in the (virtual) lenticule. In this manner the display continues showing the
20 end of the viewing range even when the viewing angle input exceeds the value that
corresponds to the end of the viewing range.

A variety of different parameters can be used to simulate the viewer angle. For example,
a detection unit that is capable of detecting a viewer and determining the viewer's
25 position relative to the detection unit may generate a viewer angle parameter. This could
for instance be implemented with a video device such as is commonly incorporated in
electronic devices and an image recognition unit such as is available in commercial
software. In a closer range eyeball tracking can be used to determine a viewer angle
parameter. Eyeball tracking systems that are commercially available can determine eye
30 position or eye movement.

Further input possibilities are available for displays that are mobile, or not fixed in
position. Numerous commonly available handheld electronic devices such as PDAs,

- 38 -

tablet computers, smartphones, games consoles, and laptops have tilt detection units incorporated. The tilt of a device relative to the horizontal plane for example can be used as an indicator of the viewer angle parameter. For example if the user tilts the display away from himself, the images he sees of the 3D data scene are the ones that would be
5 seen if the viewer had moved around the scene. Tilt can be determined by different methods and sensors, including accelerometers, gyroscopes, or front/rear/anywhere facing cameras. Similarly, handheld electronic devices often incorporate an acceleration detection unit. This type of unit can be used to detect motion of the device, which in turn can be used to determine a viewer angle parameter. For example, if a user moves the
10 device from right to left, the images he sees of the 3D data scene are the ones that would be seen if the viewer had moved from left to right. Of course the viewer angle parameter can also be selected manually by the viewer, for instance with a graphical user interface element such as a slider, or with an input device such as a mouse.

15 For reasons of clarity the preceding description and illustration have been limited to viewpoints of the 3D scene that are arranged along a line, typically a horizontal line parallel to the display. The method is easily extended for a greater spatial coverage of the viewpoints. For example, instead of a line of viewpoints, a whole rectangular area of viewpoints could be considered, and a composite representation of the full two
20 dimensional range of viewpoints produced. Here two parameters could be used to generate and display an image, for instance an x- and a y- position, or two tilt angles (e.g. one to an x-axis and the other to a y- axis). In analogy to the above-mentioned mapping between the source quad and destination pixel array, the mapping in two dimensions would determine which image corresponding to a particular view is
25 generated in the destination pixel array.

In addition to viewpoints being distributed throughout a viewing plane, as described above, the generated display image can be adapted for viewpoints nearer or further away from the display. Usually the viewpoints are assumed to be at a great distance
30 from the display, at optical infinity defined in that the viewing direction from a viewpoint is considered parallel for all pixels of the display. As discussed in the description relating to autostereoscopic displays, if the viewpoints are closer to the display, then the pixel group relating to a lenslet must be slightly wider than the lenslet itself for the viewing

- 39 -

rays to converge correctly. In the display method described here viewpoint proximity correction can be done analogously. Here, viewpoint proximity correction can be accommodated in the procedure of drawing from source quad to destination pixel array. A linear scaling transformation is applied to either the source quad (increasing the
5 relative size of the source quad) or to the destination pixel array (decreasing the relative size of the destination quad). The procedure of selection and interpolation from source quad to destination pixel array is the same as described above.

The above-described procedure of correcting the displayed composite image for a
10 particular viewing distance can be adapted in response to a viewer positioning himself in front of the display. Detection of an actual viewer position can be implemented for instance by a webcam, as is frequently built into displays. In this manner the optimum viewing distance of the display is adapted in dependence of an actual viewer position, in real-time.

15 The composite viewpoint representation data carries the source 3D information as a sampled continuous data distribution comprising intensities/colours. This is mapped to the source quad and decoded into an image for display. For the composite viewpoint representation data viewpoints may be selected arbitrarily, although spacing at regular
20 angular intervals helps maintain consistent display quality. The composite viewpoint representation data, representing a continuum of viewpoint image data, can be generated in several ways:

1. It can be produced from a number of pre-rendered viewpoint images;
2. It may be rendered from volume or multi-depth data sets using view ray casting
- 25 3. It may be generated by projecting a 3D scene through a virtual encoding element (lenslet) - referred to herein as 'forward mapping'

These methods will be described in more detail in the following. It is noteworthy that these methods are suitable for display with an autostereoscopic display, as well as for
30 the viewpoint-dependent display described above.

The composite viewpoint representation data may be produced from a set of pre-rendered viewpoint images of a 3D data scene. Pre-rendering of such viewpoint images

- 40 -

may follow any customary rendering algorithm such as ray tracing, scanline rendering or volume ray-casting, for example. Instead of pre-rendered images of a 3D data scene, a multitude of images from a multitude of viewpoints may be taken of an actual scene with a camera or other image recording device. The multitude of images may be recorded,
5 for example, with digital cameras, camcorders, or other recording devices. The multitude of pre-rendered viewpoint images may be interlaced to produce composite viewpoint representation data.

Generating the composite viewpoint representation data from a multitude of views of a
10 3D scene allows some freedom in choosing how the images are interlaced, compared to an autostereoscopic display with a pre-determined decoding element. In autostereoscopic displays the width of the interlaced image segments is determined by the lenslet width: the interlaced image segment width is equal to the lenslet width divided by the number of images from different views in the composite viewpoint representation
15 data. In methods described here, in the absence of lenslets, there is greater freedom in generating the composite viewpoint representation data.

It would be possible at one extreme to assume a single lenslet and arrange the complete images beside one another (similar to the .mpo multipicture object data format). In an
20 autostereoscopic display, as in the described simulation of the autostereoscopic display, this arrangement is not ideal, because in a viewing area that corresponds to an image boundary the viewing experience may be relatively poor. The interpolation between image sections that are not closely related to one another can produce low quality results. At the other extreme, one pixel from each image from the different selected
25 views is included behind each lenslet. In this case the composite multi-viewpoint image is the same size as when the complete images are arranged beside one another, but the arrangement of the image segments is more favourable. The image segments (one pixel wide) are as narrow as possible, and the mismatch at the image boundary is the smallest possible (without increasing image resolution and pixel numbers). In this case
30 the interpolation required to generate an image for an arbitrary viewer angle input parameter is a better approximation to the ideal, continuous-viewpoint image.

In generating the composite viewpoint representation data, the virtual lenslet is then

- 41 -

optimally of a width that is equal to the number of viewpoints multiplied by the width of an image segment (optimally: one pixel). The number of viewpoints can be chosen depending on the size limitation preferences of the composite viewpoint representation data (size increases with number of viewpoints), and the level of desired accuracy of the composite viewpoint representation data in its representation of the continuum intensity distribution.

The composite viewpoint representation data may be produced by rendering from volume or multi-depth data sets using view ray casting. In this case, the composite viewpoint representation data are generated pixel by pixel. For each pixel the following procedure is repeated:

1. Determine the viewpoint position from the current pixel;
2. Calculate the ray direction from the viewpoint through the current pixel;
3. Calculate a start in front of the data volume; and
4. Cast a ray through the volume in a stepped manner, accumulating the pixel colour value.

For an autostereoscopic display, and as described in more detail above, determining the viewpoint position from the current pixel (item 1 above) in view ray casting is dependent on the specific applied decoding element. In composite viewpoint representation data that are not related to a specific decoding element, the viewpoints could be defined prior to rendering. Alternatively, a pre-determined virtual decoding element and a pre-determined pixel array may be used.

An alternative method of generating composite viewpoint representation data is the 'forward mapping' method, wherein the 3D scene is projected through virtual lenslets of a virtual encoding element to generate the composite viewpoint representation data. This will in general generate a sampled data distribution without any inherent regularly spaced viewpoint structure. This is described in more detail as follows.

Figures 17 and 18 illustrate the core scheme of forward mapping. A source three dimensional image data structure (with scene data elements 1700) is defined, as it would be in the ray casting method described above. This is typically in the form of a data

- 42 -

volume, or more generally any sparse set of image elements 1700 at defined Z depth positions.

Rather than tracing viewing rays from destination pixel 1704 centres, through a virtual
5 encoding element (lenslet) 102, and out into the scene ('reverse' or 'view' ray casting, described in detail elsewhere), the method described here makes a direct geometric mapping, through the centre of curvature of every lenslet 102, from the boundaries of the scene data element 1700 to corresponding locations along the destination pixel plane 1702. This results in a 'filled line' 1706 existing in continuous space at the destination
10 pixel plane 1702, which is then drawn into the corresponding pixels 1704. The projection of geometry 1800 and 1802 that contributes to the filled line 1706 is shown in solid lines. Note that the mapped edges may in general not correspond with destination pixel 1704 centres (which are themselves at arbitrary locations); a sampling process takes place to determine the correct contribution to each pixel 1704 according to coverage and some
15 predefined sampling filter function.

The mapping of scene data elements 1700 imposes its own sampling function in scene space. In the illustrated example, the sampling function is zeroth-order (impulse), but better (smoother) sampling could be done using 1st order (box), or more elaborate
20 functions. In general, a sampling filter function may be imposed on scene data elements 1700 before they are mapped through the lenslets 102.

Effectively, the method of generating composite viewpoint representation data can be considered to be a 'scene data-order' version of the existing 'distribution data-order'
25 process, with sampling then taking place at the distribution surface rather than within the scene data.

Figure 17 illustrates that the effect of mapped scene data elements is limited to the field of the lenslet through which they have been mapped. That is, the width of a hit
30 distribution for a lenslet is clamped to the width of the lenslet's field p_f 1608. The projection of geometry 1800 and 1802 that is non-contributing to the filled line 1706 is shown in dashed lines. The width of the field 1708 is determined by the distance of the source aperture for the whole system, which sets the optimal viewing distance if the

- 43 -

composite viewpoint representation data is decoded using an identical lenslet array with no scaling of the distribution.

Care must be taken to ensure that possible obscuration by other scene data elements is considered (effectively elements being 'in the shadow' of other elements). If this is not done, when the final 3D image is re-displayed scene elements may be visible that should not be. The projection of geometry 1800 that is obscured by geometry 1802 and therefore is non-contributing to the filled line 1706 is shown in dotted lines.

This forward mapping method may inherently store a completely continuous viewpoint structure. This means that the parallax behaviour of the replayed scene can be optimally and perfectly smooth, with no regular discretisation into viewing zones. In this sense the method is comparable to the use of infinite source viewpoints in the reverse ray casting algorithm.

In detail, the forward mapping method proceeds as follows. First, the receiving pixel array which is to accumulate the mapped data distribution is cleared. Then, for each scene data element, the steps of the method repeated for each simulated encoding lenslet across the array are:

1. Compute the points at the hit surface (behind the lenslet) corresponding to the edges of the scene data element. For a scene data element centred at (x_p, z_p) with lateral extent w_p , the two hit points associate with simulated lenslet number i have lateral positions $x_{f1}(i)$ and $x_{f2}(i)$:

$$x_{f1}(i) = [i p_l z_p + (t-R)(x_p - w_p/2 - i p_l)] / z_p$$

$$x_{f2}(i) = [i p_l z_p + (t-R)(x_p + w_p/2 - i p_l)] / z_p$$

for lenslet parameters: p_l (width of a lenslet), t (lenslet thickness; also back focal distance) and R (lenslet radius of curvature). Note that the origin of the (x,z) coordinate system is the centre of curvature of the system's central lenslet (lenslet $i = 0$).

2. Arrange (sort) the two sort points to generate left-hand and right-hand lateral hit coordinates $x_{fL}(i)$ and $x_{fR}(i)$ respectively, such that $x_{f1}(i) \leq x_{f2}(i)$:

- 44 -

$$x_{fL}(i) = \min(x_{f1}(i), x_{f2}(i))$$

$$x_{fR}(i) = \max(x_{f1}(i), x_{f2}(i))$$

- 5 3. Clamp both sorted hit point locations so that they fall within the lenslet field. For a source aperture distance D away from the front surface of the lenslet array, each lenslet field has width p_f :

$$p_f = D p_l / [D - t + R],$$

- 10 and the left and right clamped hit point locations $x_{fLC}(i)$ and $x_{fRC}(i)$ are:

$$x_{fLC}(i) = \max(x_{fL}(i), i p_f - p_f/2)$$

$$x_{fRC}(i) = \min(x_{fR}(i), i p_f + p_f/2)$$

- 15 Hit point locations should also be tested for obscuration by closer scene data elements, and clamped to the appropriate lateral positions computed for the closer scene data elements according to the method of (1) above, if necessary.

4. Define a continuous line between the clamped hit points.

- 20 5. Sample this continuous line laterally into the receiving pixel array, and sum it with information already in the array to produce a set of accumulated values. This sampling is carried out using any appropriate filtered sampling method and function.

- 25 After the above steps have been executed for every lenslet and every scene element, the accumulated data distribution is scaled and stored ready for subsequent display and 3D replay.

- 30 Note that the above description includes only one lateral dimension (generally but not necessarily the horizontal), but the scheme is not limited to this and may for example be extended to operate over a two dimensional array of lenslets to generate a distribution encoding omnidirectional parallax. In each step, corresponding variables are computed for a second lateral direction y and lenslet number j (usually but not necessarily orthogonal to the first direction x and lenslet number i). The scene data element will

- 45 -

have vertical extent h_p .

All the above described methods for generating composite viewpoint representation data may use source 3D scene data from a variety of sources, including non-optical sources.

- 5 For instance 3D medical visualisation data from X-ray, magnetic resonance, computer tomography, or ultrasound; confocal microscopy, infrared imaging, multispectral data, or other sources may be used.

The viewpoint-dependent display described above only presents a single image that
10 corresponds to a single viewpoint. In an autostereoscopic display, the lenticular array determines the final image (generated from the composite viewpoint representation data) that is seen by each eye of the viewer. In the method described here, the sampling function determines the final image seen by both eyes of the viewer. Changing the presented image (corresponding to a different viewpoint) provides a sense of parallax to
15 the viewer, and gives an indication of the three-dimensional nature of the scene. For a full 3D spatial perception with full parallax, a stereoscopic display panel presents an image to the viewer's right eye, and a different image to the viewer's left eye. To provide stereoscopic images of the scene, the described viewpoint-dependent display method can be extended to accommodate a stereoscopic display, using for instance a display
20 with a built-in static parallax barrier or a lenslet array; or a display that requires stereoglasses; or any other autostereo- or stereo-display. Generally, the method is applied twice, to extract two viewpoint images from the composite viewpoint representation data, and then present them to the stereoscopic display system for onward stereoscopic display to the user using whatever left/right multiplexing means it employs. This is
25 described in more detail in the following section.

The basic tilt-dependent display unit described hitherto uses for example a conventional flat panel display component, such that an identical image is provided to each eye of the viewer. The interactive extraction of geometrically correct viewpoints from a stored
30 continuum of such viewpoints according to viewer/display angle leads to effective stimulation of the motion disparity depth cue in the human visual system, and this effect is largely responsible for the highly effective three dimensional capability of the unit.

- 46 -

A yet more powerful three dimensional experience is gained if the above basic scheme is combined with a unit which has a physical stereoscopic (or preferably autostereoscopic) display component; this can augment the display system with the binocular disparity depth cue, providing a different image to each eye in the correct
5 spatial relationship.

Many methods exist for multiplexing a stereo image pair for a display panel and then demultiplexing them again before they reach the viewer's eyes. These are broadly classified as:

10

1. *Stereoscopic (or non-autostereoscopic)*: the viewer must wear special apparatus (e.g. glasses) to demultiplex the stereo image pair;

15

2. *Autostereoscopic*: demultiplexing of left and right views to left and right eyes takes place within the display unit itself, and no glasses are needed.

In either case, the first requirement is that now two viewpoints rather than one are generated from the composite viewpoint representation data; each of the two viewpoints corresponds to a view of the original scene which would be seen by the respective eye
20 of the viewer. Meta-information about the viewing geometry for the stored scene should be stored with the composite viewpoint representation data to enable the display unit to compute the viewpoint angles correctly.

A second requirement applies only in the case of autostereoscopic display panels,
25 because they enforce a specific geometric relationship between themselves and the viewer's eyes. Non-autostereoscopic displays have no inherent viewer angle dependence – the same stereo image pair is available wherever the viewer moves within the overall field of view of the display. To support non-autostereoscopic display units in a full parallax interactive display system, no special processing or treatment is required
30 beyond that which is needed to provide the left/right image multiplex at the display panel component, and would be needed to support a standard, fixed viewpoint stereoscopic image using the same equipment (typically time-multiplexed using active shutter glasses, or spatially multiplexed using polarising filters, using passive (depolarising) glasses).

- 47 -

Autostereoscopic displays, however, inherently produce images which vary with viewer/display angle – they must do this to obviate the need for eyewear in the stereo demultiplexing process. The left and right images of the stereo pair are projected into
5 angular zones in front of the display screen through the use of a direction selective element, the two most common being the static parallax barrier and the lenslet array (lenticular sheet being one example of the lenslet array arrangement). Because the relationship between the displayed left/right image pixels and the direction selective element is fixed, the left/right image projection zones are fixed relative to the display
10 panel, and the panel user must remain in the correct location, perpendicular to the display panel, to receive the correct image at each eye. If the display equipment is tilted in the tilt-dependent display described above, the correct perception of the stereo pair is lost, even when the display is moved only a small angle from the normal.

15 The effect identified above may be compensated by incorporating an extra processing step for use with fixed native autostereoscopic display components. In this tilt compensation step, the displayed image information is transformed. This may be achieved, for example, with a lateral shift transformation in dependence upon the tilt angle of the unit. With this transformation left and right image pixels remain projected to
20 the correct zones relative to the viewer. Figure 19 illustrates the tilt compensation in the case of a static parallax barrier 1900, and Figure 20 illustrates tilt compensation in the case of a lenslet array 2000. The display pixel surface 1904 displays image information for the viewer's left eye 1906 and the viewer's right eye 1908. The parallax barrier 1900 with the barrier slits 1902 (respectively the lenslet array 2000) ensures that the correct
25 image is seen by each eye. To compensate for tilt of the display by an angle α 1910, a pixel data shift 1912 is performed in the tilt compensation step.

Implementation of the tilt compensation step incurs a negligible performance penalty, as the required image shift can be integrated into the transfer of samples from the source
30 distribution to the final rendered display. For example, the tilt compensation step may occur after the display image is generated. The tilt compensation step may also be integrated into the algorithm that generates the display image.

- 48 -

Expressed as a proportion of the total field width p_f associated with a lenslet or barrier
5 slit, the translation s_c 1912 required is:

$$s_c = -d \tan \alpha / p_f$$

for a device rotation angle α , with d denoting the distance between the pixel plane 1904
10 and the parallax barrier 1900 (respectively between the pixel plane 1904 and the lenslet
centre of curvature: $d = t - R$, with t being lenslet back focal distance from pole, and R
being the radius of curvature of the lenslet).

The display described above may provide a three dimensional display unit using
15 interactive motion disparity and binocular disparity to produce a highly engaging and
realistic display with full parallax features, using commercially available mobile display
devices and components.

The described concept is based on the recognition of there being a relationship between
20 the discrete images; discrete images provide 'point angular samples' of the space
(angular impulse sampling). On the basis of this recognition information on pixels (and
hence views) that lie between the discrete images can be deduced. This is distinct from
an approach where there is no known (a priori) relationship, and no information on pixels
that lie between their source images; in this approach an estimate of the underlying 3D
25 structure must be made in order to generate new viewpoints. To estimate the 3D
structure from 2D image pixels it is then necessary to find correspondence, that is,
identify points in the 3D scene that might correspond in view images. Finding
correspondence is a difficult (and underdetermined) problem, due to occlusion problems,
changes in lighting with angle, etc. Further, difficult mapping or estimating of object
30 structure may be required. For example, mesh-based models as an estimate of the 3D
structure of the scene may be generated, and then a new view of the model generated.
In morphing/warping mesh points are manipulated to create new view. While these
approaches fundamentally require information regarding the relationship between 3D

- 49 -

object space and 2D images from cameras capturing views of it, the concept described here stays entirely within image (pixel) space, and requires no further information regarding the original 3D scene.

5 The concept described here derives an in-between view image pixel using filtered resampling of the angular distribution recorded behind a [virtual] lenslet. This is based on the recognition that approximation of the 3D object structure between same pixels from different viewpoints can be performed by reconstructing a continuous intensity/colour function between them and resampling it at the appropriate point. This is valid because
10 the same pixel from different viewpoints can be treated as an area sample of an angular chunk of space. The approach requires that producing the data defining a viewpoint distribution be band-limited. If the data defining a viewpoint distribution is produced using the 'forward mapping' approach described above, then the recorded distribution is inherently and directly a set of perfectly contiguous area samples of a continuous
15 distribution. If the data defining a viewpoint distribution is produced using a different approach, for example interlacing, then enough views are provided such that the largest change between views is still recoverable (i.e. angular 3D scene image information is within the Nyquist limit - sampling rate defined by the no. of views). As the band-limited continuous intensity function is imposed by the image creation process, and hence fully
20 known, linear resampling of the continuous intensity function produced the required new image.

Figure 21 illustrates a file format that facilitates use of the method of generating an image as described above. As explained above, the composite viewpoint representation
25 data 1201 comprises view data from a continuous viewpoint range, and as such offers a representative sample of a continuous viewpoint representation. The composite viewpoint representation data 1201 is preferably stored in a dedicated file format 2100 that usefully includes further information that is pertinent to the composite viewpoint representation data 1201. Usefully, this would include metadata 2102 about the
30 composite viewpoint representation data, for instance: the lenslet width or decoding element width for which the composite viewpoint representation data is intended; the lenslet radius of curvature, and the lenslet thickness (for calculation of a relative displacement in dependence upon a viewer angle as described above); information

describing the selection of views that represent the continuous viewpoint range; the number of images that are interlaced in the composite viewpoint representation data; information describing the viewing positions to which images relate (angular direction, coordinates, or other); or further data that relates to the composite viewpoint representation data or more generally to the pre-rendered representation of the 3D scene. Further, compression of the data may be desirable to reduce file sizes. Compression parameters may be included as metadata 2104 to the file format.

The dedicated file format described above does not necessarily have to be displayed using a method of generating and displaying an image from the composite viewpoint representation data as described in the forgoing description. The same file format can be used for autostereoscopic displays, for example. The file format can provide data for – for example – printed 3D large-format or high-volume, small-format display panels, specialised visualisation monitors, 3DTV, or other stereo/autostereo 3D display units.

Figure 22 shows a process diagram with the main steps undertaken to generate a display image for a viewpoint-dependent display. Starting with composite viewpoint representation data 1201, a display image is generated 2204 in dependence on an input 2202. The generated display image is outputted 2206 to a display 2208. For generating the display image 2204, the composite viewpoint representation data 1201 is sampled with a sampling function 2216 in dependence on an input 2202. One way of achieving this is by drawing the composite viewpoint representation data 1201 onto a source quad 2210. In dependence on the input 2202, the position (or mapping) of a destination pixel array is determined 2212. Then the source quad is drawn onto the destination pixel array 2214 according to the sampling function 2216, therewith producing the display image.

Figure 23 illustrates a typical graphics pipeline, such as may be used for generating a display image as described above. Quads are primitives in the geometry path 2322, the (source) composite viewpoint representation is a pixel array that is mapped to the quad and is stored in the texture memory 2318, and the destination pixel array is produced as the result of rasterisation 2310. The sampling, or filtering, or interpolation that generates the display image occurs in the process step of rasterisation 2310.

Figure 24 shows major components of a device utilising the process described above. The input 2202 is generated from a tilt sensor 2400, a camera 2402, or another source as discussed above. The display image is generated by a processing unit 2404 that has
5 a storage unit 2408 and a unit that is capable of processing graphical data, such as a GPU 2404. The GPU (which may be integrated into a main CPU or may be discrete) comprises a texture engine 2406 that is especially useful for generating and manipulating the source quad and destination pixel array.

10 However, the invention may alternatively be implemented on a general purpose processor (e.g. in a single processor device) or using any other suitable processing resources of the device. The processing unit may provide further functionality, for instance converting an input into a viewpoint parameter for use in generating the display image.

15

While the invention has been described in reference to its preferred embodiments, it is to be understood that the words which have been used are words of description rather than limitation and that changes may be made to the invention without departing from its scope as defined by the appended claims.

20

Each feature disclosed in the description, and (where appropriate) the claims and drawings may be provided independently or in any appropriate combination.

Claims

1. A method of providing a display of a three-dimensional scene, comprising:
receiving an input representative of a given viewpoint;
5 in dependence on the input, displaying an image providing a view of the three-dimensional scene corresponding to the given viewpoint, the displaying comprising:
accessing data defining a viewpoint distribution representing a plurality of views
of the three-dimensional scene;
sampling the viewpoint distribution in dependence on the input to generate a view
10 of the three-dimensional scene corresponding to the given viewpoint; and
displaying an image based on the generated view.
2. A method according to claim 1, comprising continuously updating the displayed
image in response to changes in the input representative of the given viewpoint.
15
3. A method according to claim 1 or 2, wherein sampling the viewpoint distribution
comprises convolving the viewpoint distribution with a sampling function.
4. A method according to any of the preceding claims, wherein the sampling step
20 comprises sampling the viewpoint distribution in dependence on one or more viewpoint
parameters determined based on the input.
5. A method according to any of the preceding claims, wherein the one or more
viewpoint parameters comprise one or more of: viewing angle information; viewing
25 distance information; and viewpoint coordinate information.
6. A method according to any of the preceding claims, wherein the viewpoint
distribution represents a continuously varying range of views of the three-dimensional
scene.
30
7. A method according to any of the preceding claims, wherein the viewpoint
distribution represents a one-dimensional range of viewpoints.

- 53 -

8. A method according to any of the preceding claims, wherein the viewpoint distribution represents a two-dimensional range of viewpoints.
- 5 9. A method according to any of the preceding claims, wherein the viewpoint distribution comprises a distribution of image data values.
10. A method according to any of the preceding claims, wherein the data comprises data samples defining the viewpoint distribution.
- 10 11. A method according to claim 10, wherein the data samples comprise image data samples and wherein the sampling step comprises resampling the image data samples.
12. A method according to claim 11, wherein the image data samples correspond to a plurality of pre-rendered views of the scene.
- 15 13. A method according to any of the preceding claims, wherein the data comprises a plurality of pre-rendered views of the scene.
- 20 14. A method according to claim 12 or 13, wherein the views are interlaced to form a composite, interlaced image.
15. A method according to any of claims 12 to 14, wherein the sampling step comprises performing an interpolation based on at least two of the pre-rendered views.
- 25 16. A method according to any of the preceding claims, wherein the method comprising:
- generating a source pixel array from the data;
 - generating a destination pixel array based on the source pixel array, the
- 30 generating step including performing a resampling operation in relation to the source pixel array; and
- generating an output image based on the destination pixel array.

17. A method according to any of the preceding claims, wherein the sampling is performed using a texture mapping engine, preferably forming part of a graphics processing unit (GPU).

5 18. A method according to any of the preceding claims, wherein the input comprises one or more of: device orientation or motion information; viewer position or motion information; eye position information; and eye movement information.

10 19. A method according to any of the preceding claims, wherein the method is performed at a device having an image sensor, and wherein receiving an input comprises receiving image information from the image sensor; and analysing the image information to determine in relation to a viewer of the device one or more of: viewer position information; viewer motion information; eye position information; and eye movement information.

15

20. A method according to any of the preceding claims, wherein the input comprises information relating to the distance between a display and a viewer, the method further comprising scaling the sampled data in dependence upon the input.

20 21. A method according to any of the preceding claims, comprising generating a plurality of images corresponding to respective different viewpoints based on the viewpoint distribution, and outputting the plurality of images.

25 22. A method according to claim 21, comprising displaying the plurality of images using a three-dimensional display system, wherein the plurality of images comprises first and second images for provision respectively to left and right eyes of a viewer.

23. A method according to claim 21 or 22, wherein respective separate viewpoints are determined for the left and right eyes based on the input.

30

24. A method of generating a multi-viewpoint representation of a three-dimensional scene, comprising:

calculating a two-dimensional projection of the three-dimensional scene through

- 55 -

at least one virtual encoding element onto a sampling plane; and
sampling the projection in the sampling plane to generate sampled pixel data;
wherein the multi-viewpoint representation comprises the sampled pixel data.

- 5 25. A method according to claim 24, wherein the calculating step comprises, for each of a plurality of scene elements of the three-dimensional scene, calculating a projection of the scene element onto the sampling plane through the virtual encoding element.
26. A method according to claim 25, wherein the projection is calculated taking
10 account of possible obscuration by other scene elements.
27. A method according to claim 25 or 26, wherein the sampling comprises determining a contribution of the projected scene element to one or more pixel samples.
- 15 28. A method according to any of claims 24 to 27, wherein the sampling step comprises accumulating pixel sample values based on pixel sample contributions determined for each of a plurality of scene elements.
29. A method according to any of claims 24 to 28, comprising performing the
20 calculating step for each of a plurality of virtual encoding elements.
30. A method according to claim 29, wherein the sampling step comprises accumulating pixel sample values based on pixel sample contributions determined for each of the virtual encoding elements.
- 25 31. A method according to claim 29 or 30, wherein each virtual encoding element is associated with a projection field in the sampling plane, and wherein the projection of scene elements through a virtual encoding element contributes only to pixel samples within its associated projection field.
- 30 32. A method according to any of claims 24 to 31, wherein the projection is calculated based on optical characteristics of the one or more virtual encoding elements.

- 56 -

33. A method according to any of claims 24 to 32, wherein the one or more virtual encoding elements comprise one or more virtual lenslets.
34. A non-transitory computer-readable medium encoding a data structure storing a multi-viewpoint representation of a three-dimensional scene, wherein the multi-viewpoint representation comprises data defining a viewpoint distribution representing a range of views of the three-dimensional scene.
35. A computer-readable medium according to claim 34, wherein the viewpoint distribution comprises a distribution of image data values.
36. A computer-readable medium according to claim 34 or 35, wherein the multi-viewpoint representation comprises image data samples defining the viewpoint distribution.
37. A computer-readable medium according to claim 36, wherein the image data samples correspond to a plurality of pre-rendered views of the scene.
38. A computer-readable medium according to any of claims 34 to 37, wherein the multi-viewpoint representation is arranged to enable generation of arbitrary views of the three-dimensional scene within the range of views represented by the viewpoint distribution.
39. A computer-readable medium according to any of claims 34 to 38, wherein the multi-viewpoint representation is compressed.
40. A computer-readable medium according to any of claims 34 to 39, the data structure further comprising metadata relating to the multi-viewpoint representation.
41. A computer-readable medium according to Claim 40, the metadata comprising at least one of: autostereoscopic lenslet width; autostereoscopic lenslet radius of curvature; autostereoscopic lenslet thickness; number of viewing window samples; viewing window definition; and compression parameters.

42. A system for providing an image of a three-dimensional scene, comprising:
means for receiving an input representative of a given viewpoint;
means for, in dependence on the input, outputting an image providing a view of
5 the three-dimensional scene corresponding to the given viewpoint, the outputting means
comprising:
means for accessing data defining a viewpoint distribution representing a plurality
of views of the three-dimensional scene;
means for sampling the viewpoint distribution in dependence on the input to
10 generate a view of the three-dimensional scene corresponding to the given viewpoint;
and
means for outputting an image based on the generated view.

43. A system for generating a multi-viewpoint representation of a three-dimensional
15 scene, comprising:
means for calculating a two-dimensional projection of the three-dimensional
scene through at least one virtual encoding element onto a sampling plane; and
means for sampling the projection in the sampling plane to generate sampled
pixel data;
20 wherein the composite multi-viewpoint representation comprises the sampled
pixel data.

44. Apparatus having means for performing a method according to any of claims 1 to
33.

25

45. A computer-readable medium comprising software code adapted, when executed
on a data processing apparatus, to perform a method according to any of claims 1 to 33.

46. A method or system for providing an image of a three-dimensional scene
30 substantially as described herein with reference to and/or as illustrated in the
accompanying drawings.

47. A method or system for generating a multi-viewpoint representation of a three-

- 58 -

dimensional scene substantially as described herein with reference to and/or as illustrated in the accompanying drawings.

48. A computer-readable medium or data structure for storing a multi-viewpoint
5 representation of a three-dimensional scene substantially as described herein with
reference to and/or as illustrated in the accompanying drawings.

49. A method of rendering images for autostereoscopic display, comprising:
generating a pixel value for each pixel of an image, each pixel being mapped to a
10 single viewpoint, such that an integer number of pixel values are generated for at least
one element of a decoder array;
storing each generated pixel value;
resampling the stored pixel values in dependence on the number of display pixels
for each element of the decoder array; and
15 displaying the resampled pixel values.

50. A method according to Claim 49, wherein the integer number of pixel values is
chosen to be greater than the number of display pixels for the at least one element of the
decoder array.

20

51. A method according to Claim 50, wherein the number of display pixels is rounded
up to the nearest integer.

52. A method according to Claim 50 or 51, wherein the number of display pixels is
25 rounded up to an integer number 1, 2, 3, 4, or 5 more than the number of display pixels.

53. A method according to any of Claims 49 to 52, wherein the view point is mapped
utilising ray tracing.

30 54. A method according to Claim 53, wherein the pixel value is generated by
integrating along the traced ray.

55. A method according to Claim 54, wherein the integration scheme utilised is one

of: emission absorption model; maximum intensity projection; and isosurface detection.

56. A method according to any of Claims 53 to 55, wherein a ray direction vector is computed directly, and in one arithmetic operation, from the view point for the each
5 display pixel.

57. A method according to any of Claims 53 to 56, wherein the ray is traced from a point a distance in front of the display to a point a distance behind the display, such that 3D object space extends in front of and behind the display surface.

10

58. A method according to any of Claims 53 to 57, further comprising determining unused pixels utilising a masking operation.

59. A method according to Claim 58, wherein the masking operation uses data
15 volume convex hull projection.

60. A method according to Claim 58 or 59, wherein the pixel values are not generated for the unused pixels.

20 61. A method according to any of Claims 49 to 60, further comprising determining an intensity distribution from the generated pixel values, wherein the resampling utilises said intensity distribution to determine said resampled pixel values.

62. A method according to Claim 61, wherein said intensity distribution is determined
25 utilising linear interpolation.

63. A method according to Claim 61, wherein said intensity distribution is determined utilising a fitted polynomial curve.

30 64. A method according to any of Claims 49 to 63, further comprising performing a transformation on the generated pixels before resampling.

65. A method according to Claim 64, wherein said transformation is at least one of:

linear scaling; linear shift; translation; and other affine.

66. A method according to Claim 64 or 65, wherein transformation is performed in dependence on at least one physical parameter of the decoder array.

5

67. A method according to any of Claims 49 to 66, wherein resampling is performed in dependence on at least one physical parameter of the decoder array.

68. A method according to Claim 66 or 67, wherein the at least one physical parameter is at least one of: temperature; stress; and deformation.

10

69. A method according to any of Claims 64 or 65, wherein transformation is performed in dependence on at least one environmental parameter.

70. A method according to any of Claims 49 to 69, wherein resampling is performed in dependence on at least one environmental parameter.

15

71. A method according to Claim 69 or 70, wherein the at least one environmental parameter is at least one of: humidity; and temperature.

20

72. A method according to any of Claims 49 to 71, wherein the decoder is a micro-optical array comprising a plurality of lenslet elements or a parallax barrier.

73. A method according to Claim 72, wherein the plurality of lenslet elements are substantially cylindrical or rotationally symmetrical.

25

74. A method according to any of Claims 49 to 73, wherein the resampling utilises a graphical processing unit.

75. A method according to any of Claims 49 to 74, wherein the resampled pixels are printed.

30

76. A method according to any of Claims 49 to 75, wherein the resampled pixels are

- 61 -

displayed electronically.

77. A method according to any of Claims 49 to 76, further comprising:
computing the viewpoint position from the stored pixel position for the decoder
5 array;
calculating a ray direction from the viewpoint through the stored pixel;
moving back along the ray vector from the stored pixel by a single data volume
side length;
casting the ray through the volume for a multiple of volume side lengths, in a
10 stepped manner, to accumulate the pixel colour value; and
adjusting the image and viewing distance parameters for display.
78. A method according to Claim 77, wherein the multiple is $4\sqrt{3}$.
- 15 79. A method according to any of Claims 49 to 78, wherein display data is subjected
to affine transformation by transforming data lookup coordinates.
80. An apparatus for rendering images for autostereoscopic display, comprising:
means for generating a pixel value for each pixel of an image, each pixel being
20 mapped to a single viewpoint, such that an integer number of pixel values are generated
for each element of a decoder array;
means for storing each generated pixel value;
means for resampling the stored pixel values in dependence on the number of
display pixels for each element of the decoder array; and
25 a display that displays the resampled pixel values.
81. A system for rendering images for autostereoscopic display, wherein:
a pixel value is generated for each pixel of an image, each pixel being mapped to
a single viewpoint, such that an integer number of pixel values are generated for each
30 element of a decoder array;
each generated pixel value is stored;
the stored pixel values are resampled in dependence on the number of display
pixels for each element of the decoder array; and

- 62 -

the resampled pixel values are displayed.

82. A method of rendering autostereoscopic images, comprising:
generating a pixel value for each pixel of an image, each pixel being mapped to a
5 respective viewpoint; and
printing the generated pixel values.
83. A method for rendering images for autostereoscopic display substantially as
herein described with reference to and/or as illustrated in the accompanying drawings.
10
84. An apparatus for rendering images for autostereoscopic display substantially as
herein described with reference to and/or as illustrated in the accompanying drawings.
85. A computer program product for rendering images for autostereoscopic display
15 substantially as herein described with reference to and/or as illustrated in the
accompanying drawings.
86. A system for rendering images for autostereoscopic display substantially as
herein described with reference to and/or as illustrated in the accompanying drawings.

1 / 16

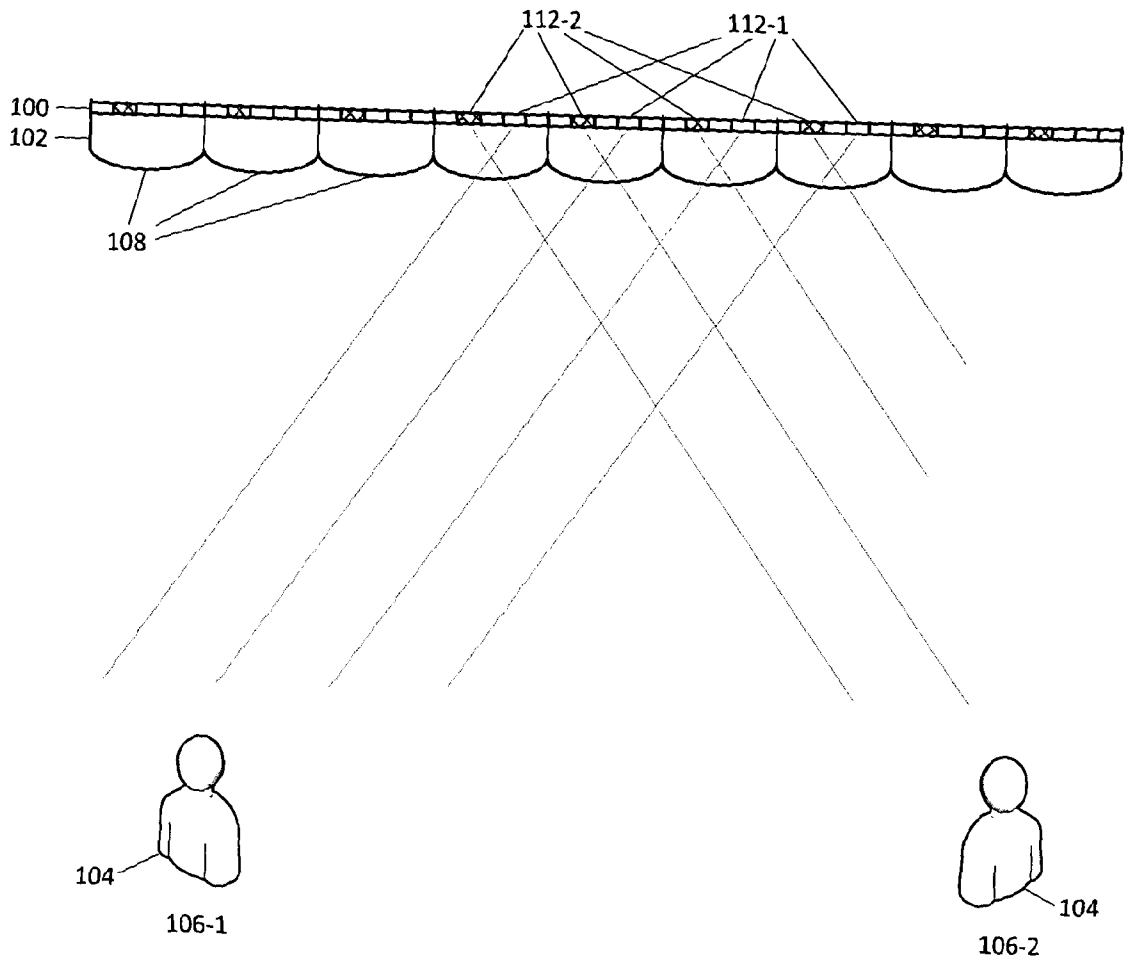


Figure 1

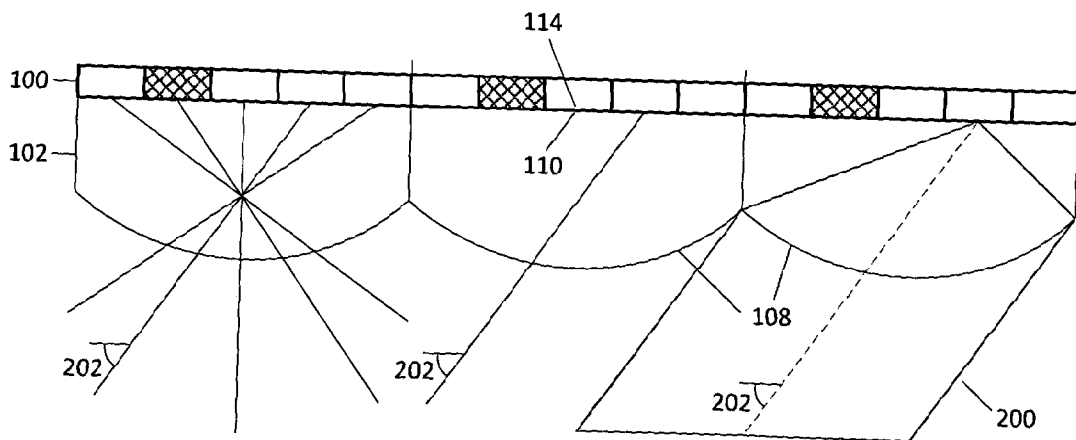


Figure 2

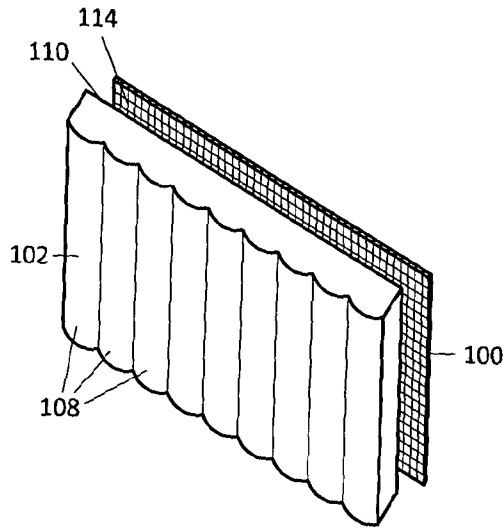


Figure 3

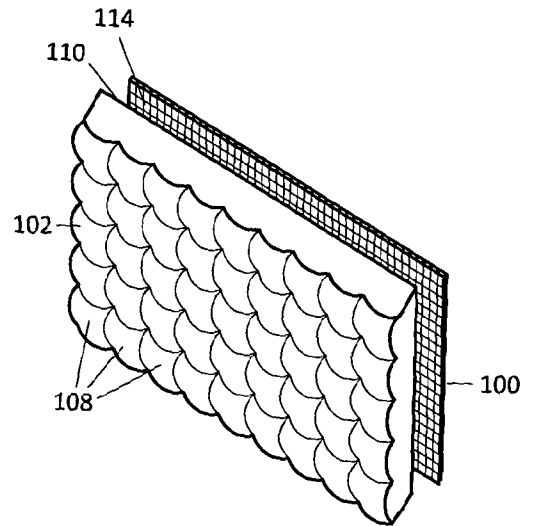


Figure 4

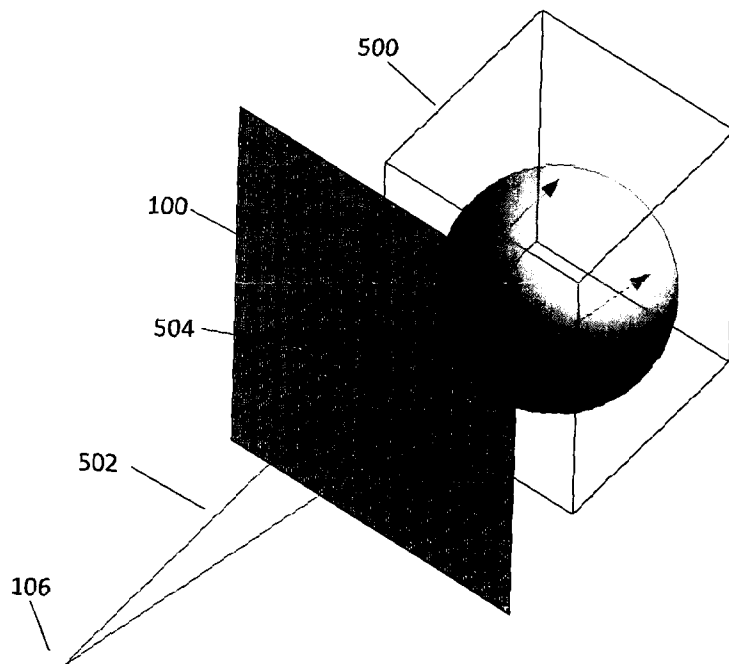


Figure 5

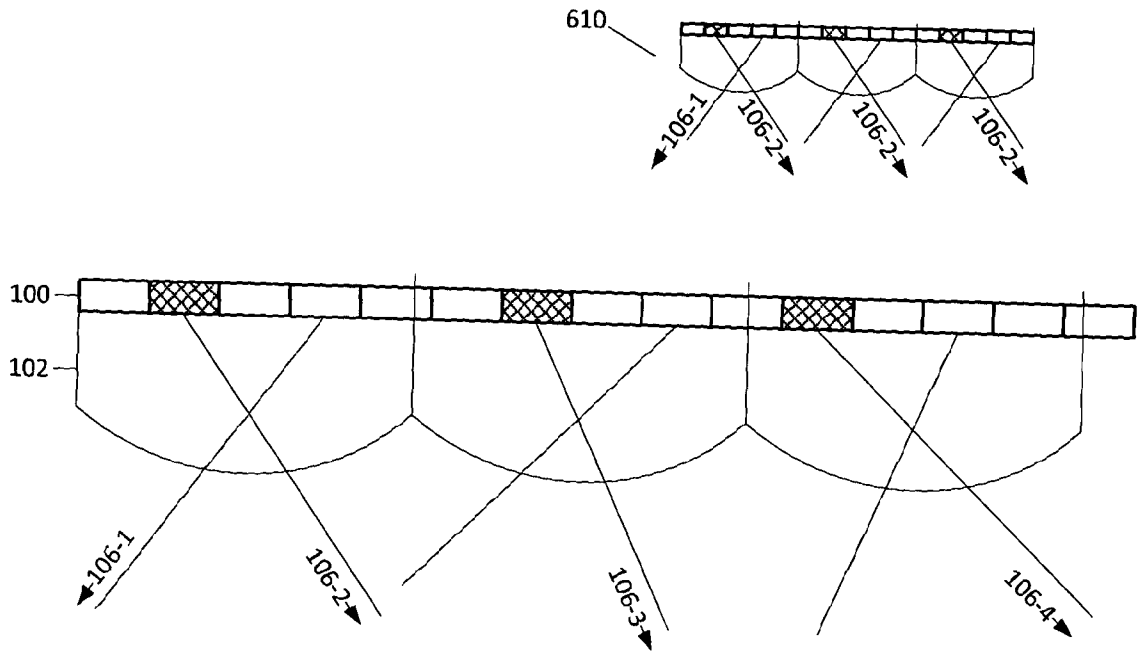


Figure 6

4 / 16

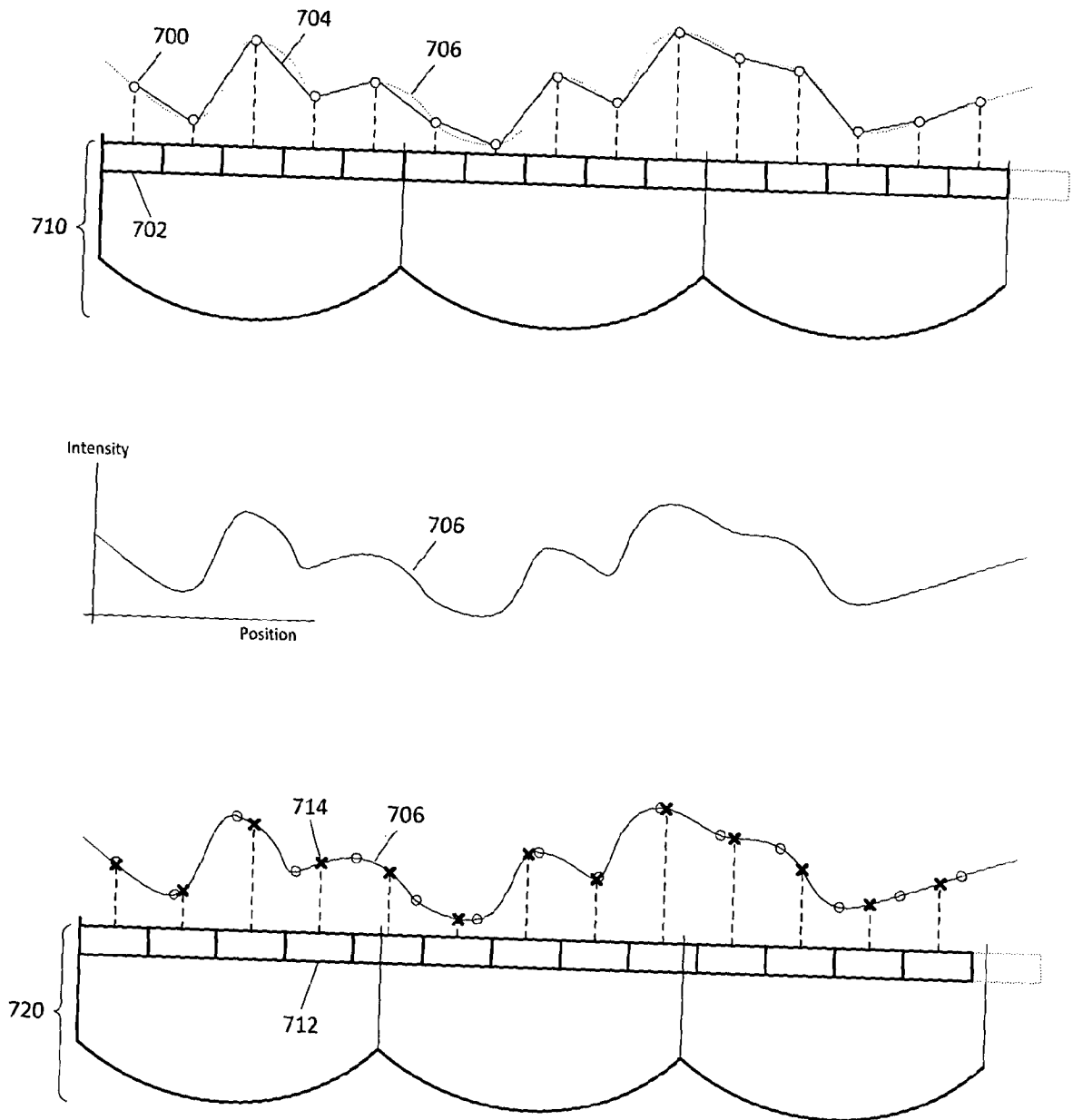


Figure 7

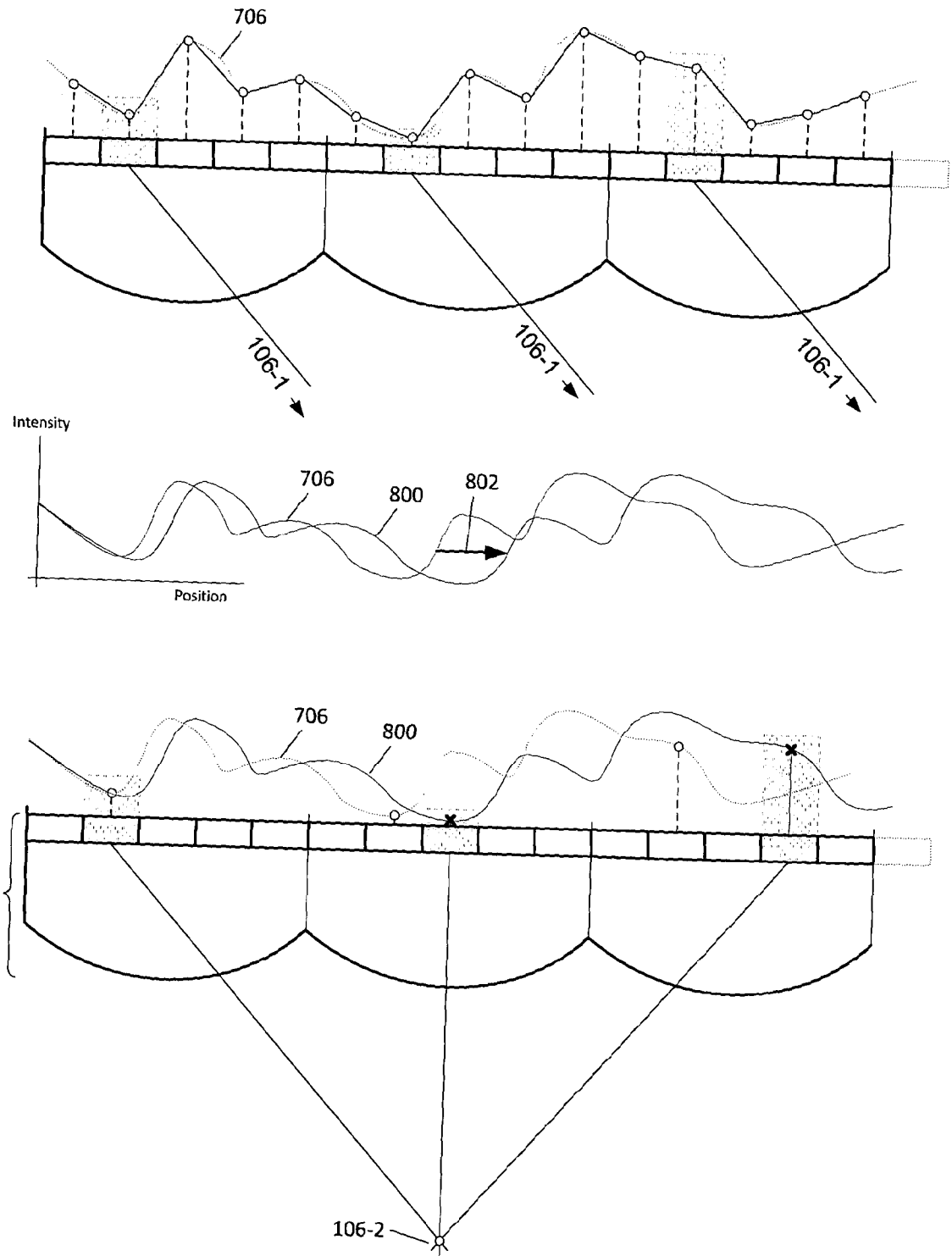


Figure 8

6 / 16

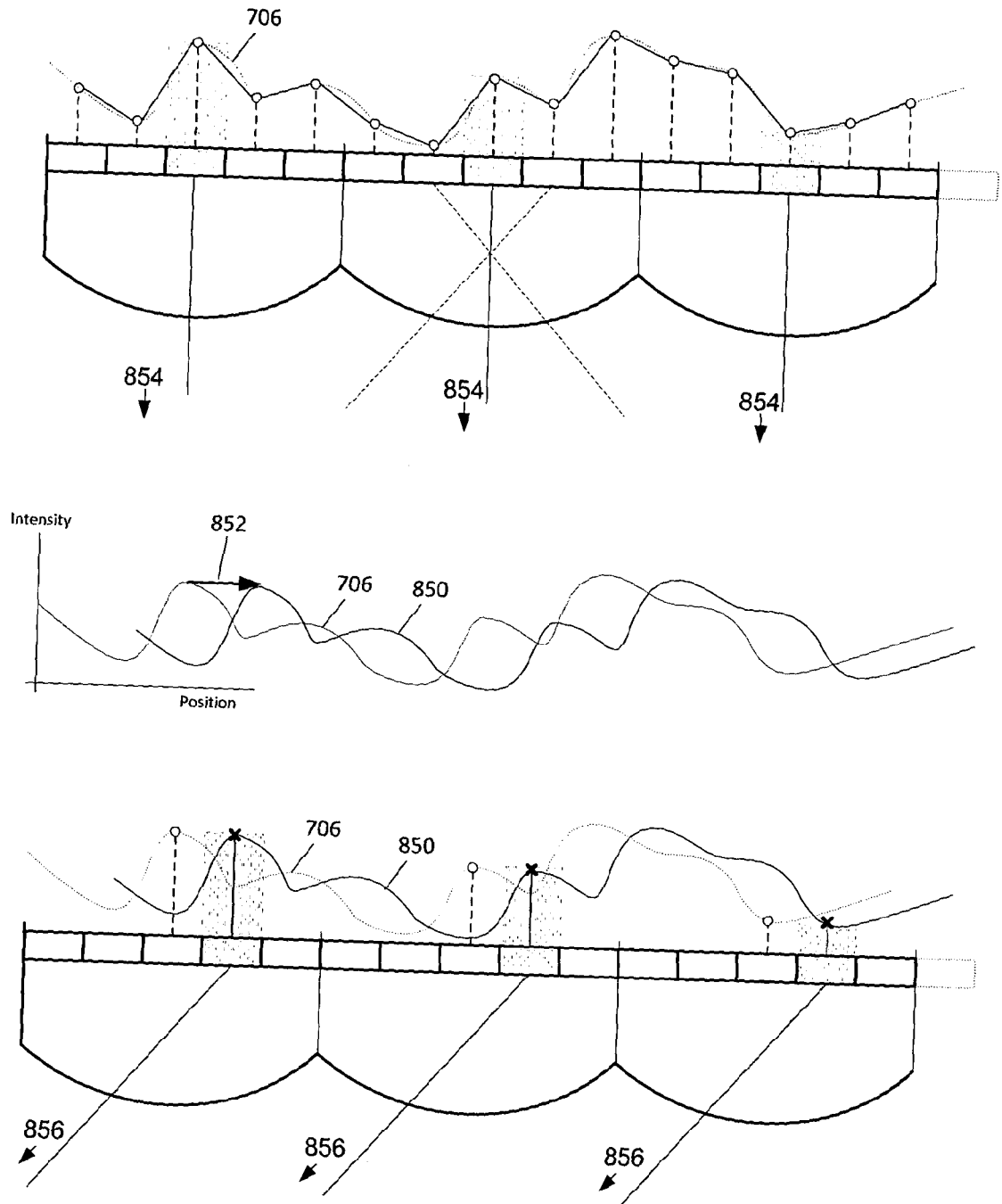


Figure 9

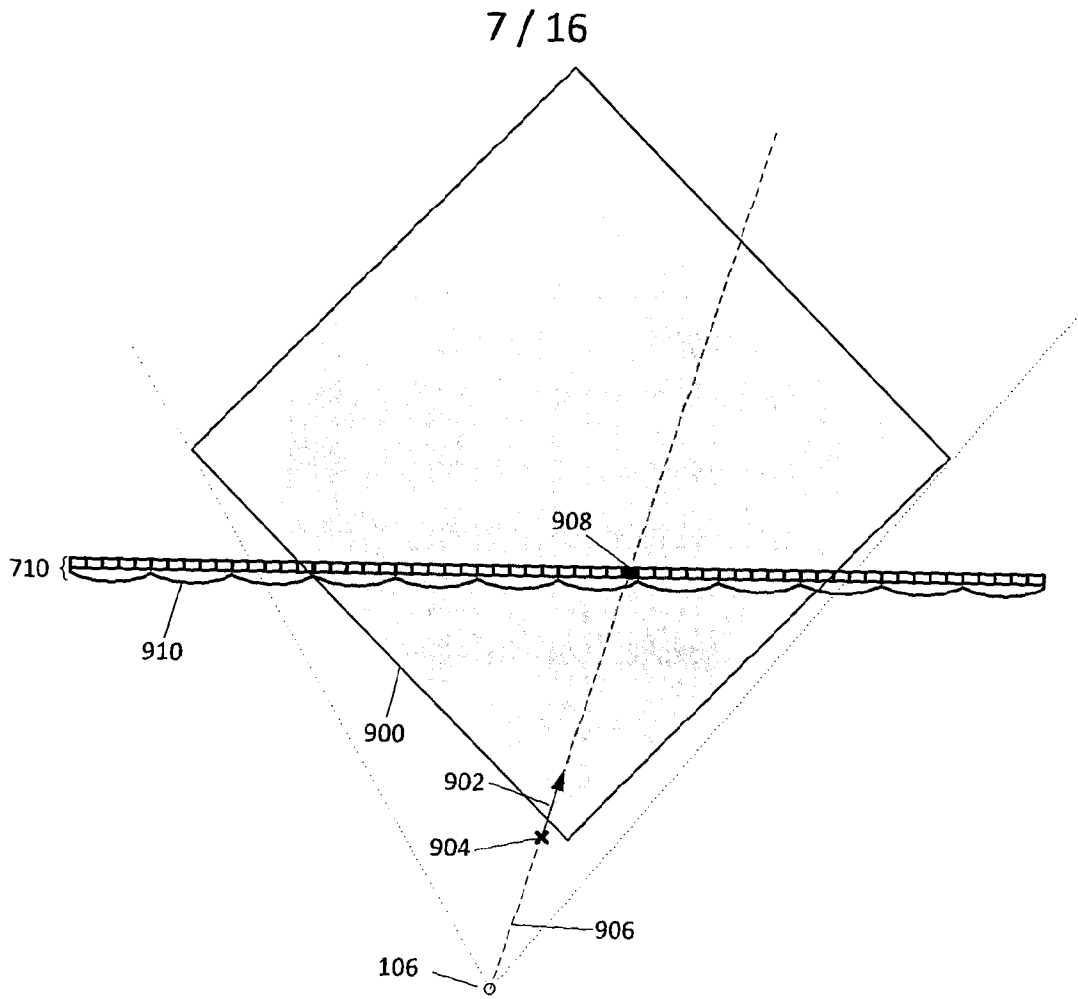


Figure 10

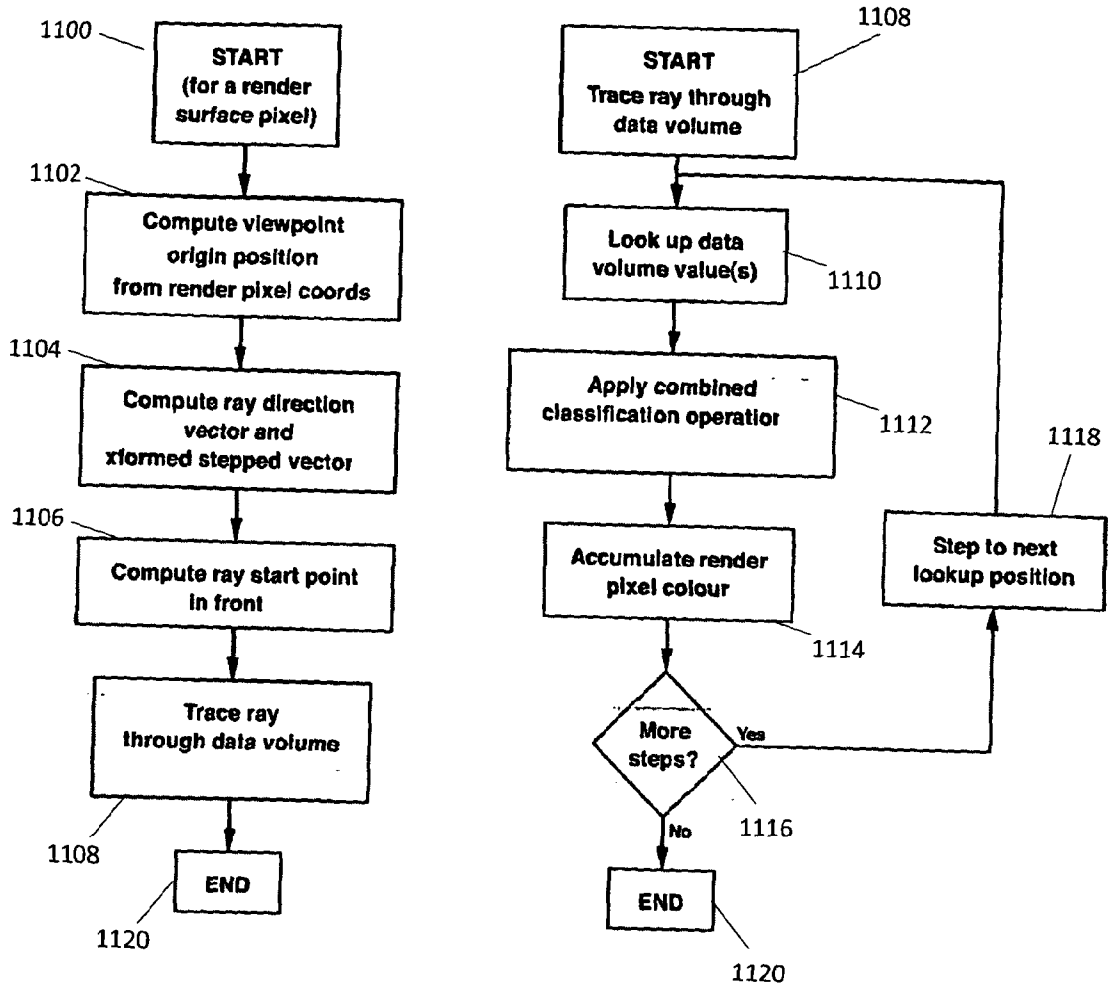


Figure 11

9 / 16

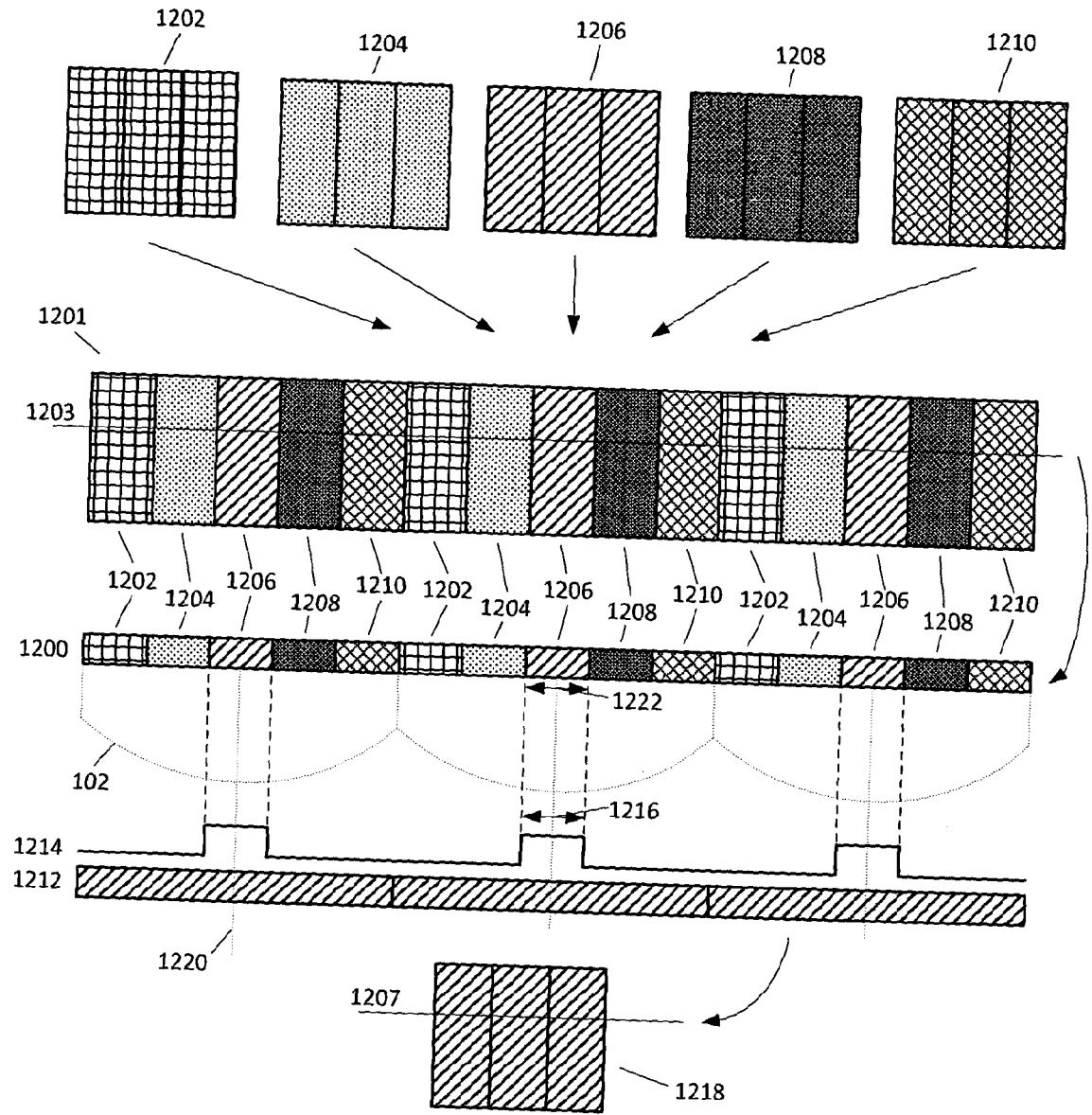


Figure 12

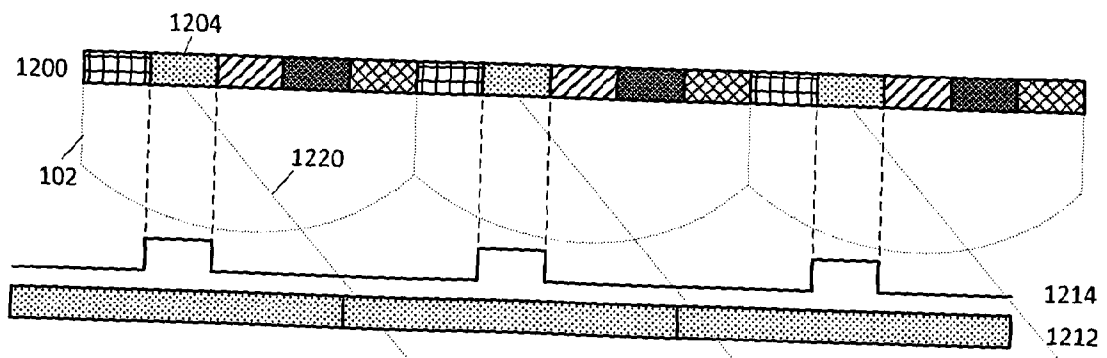


Figure 13

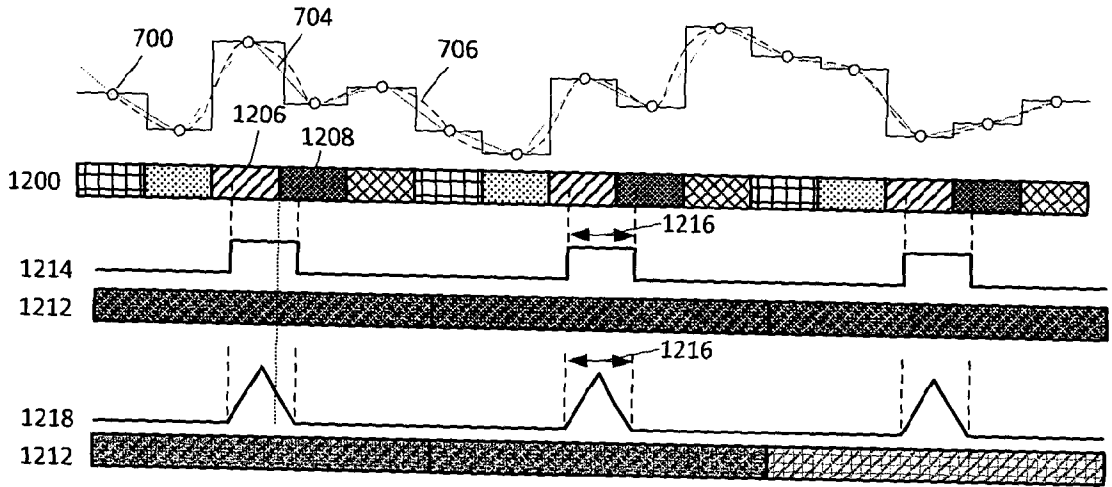


Figure 14

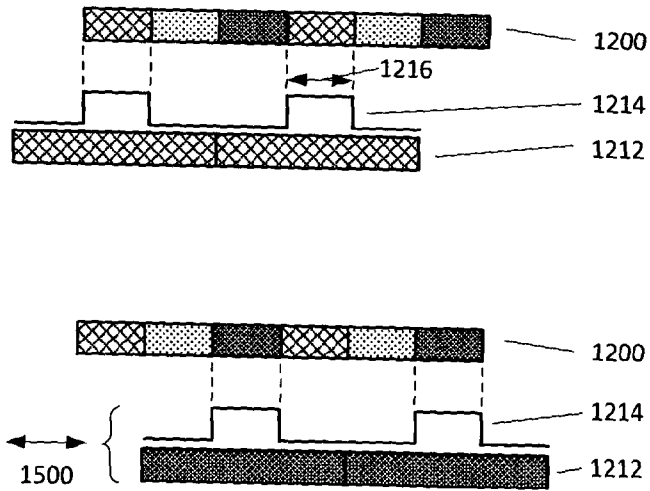


Figure 15

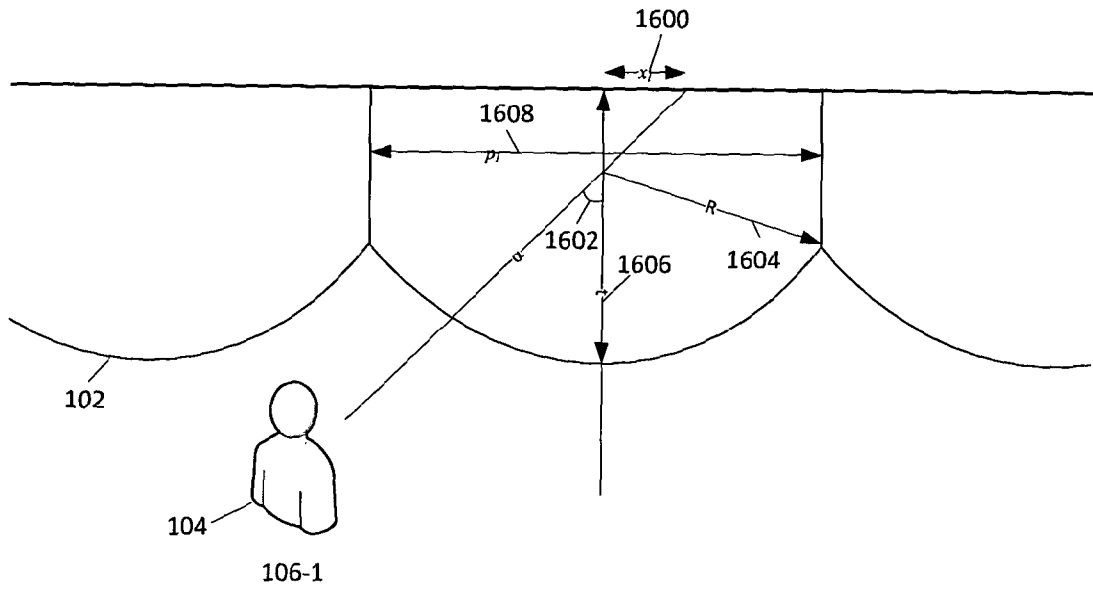


Figure 16

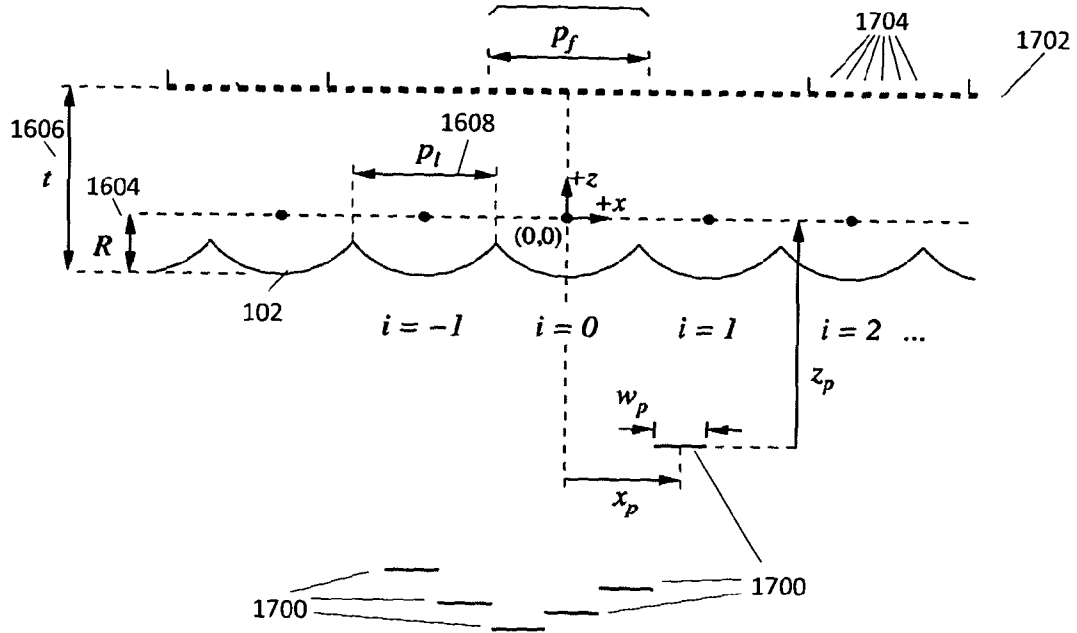


Figure 17

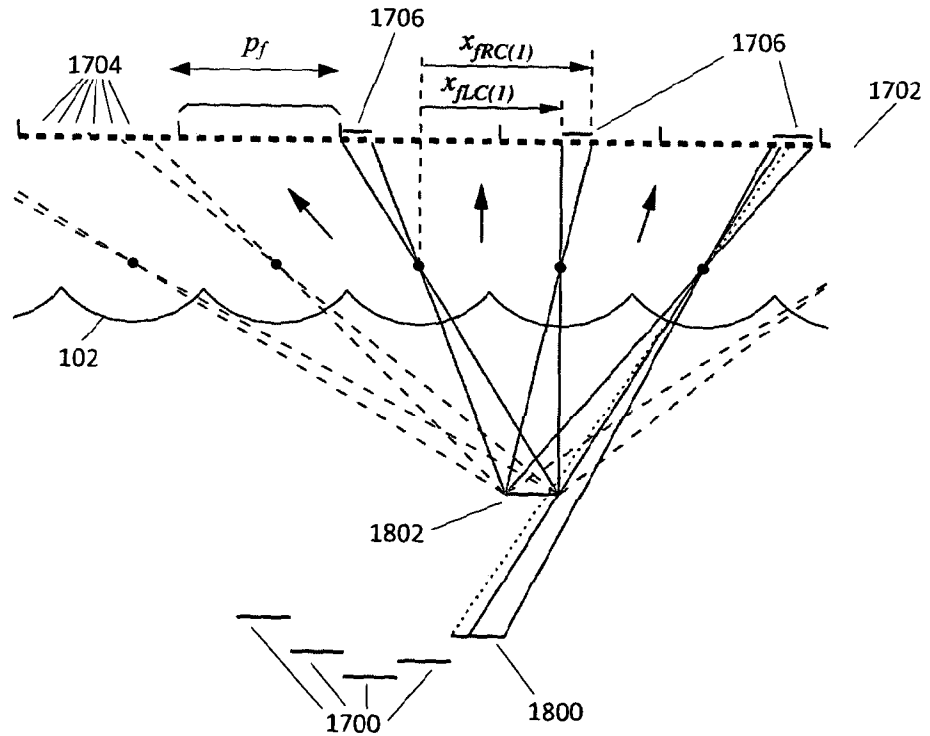


Figure 18

13 / 16

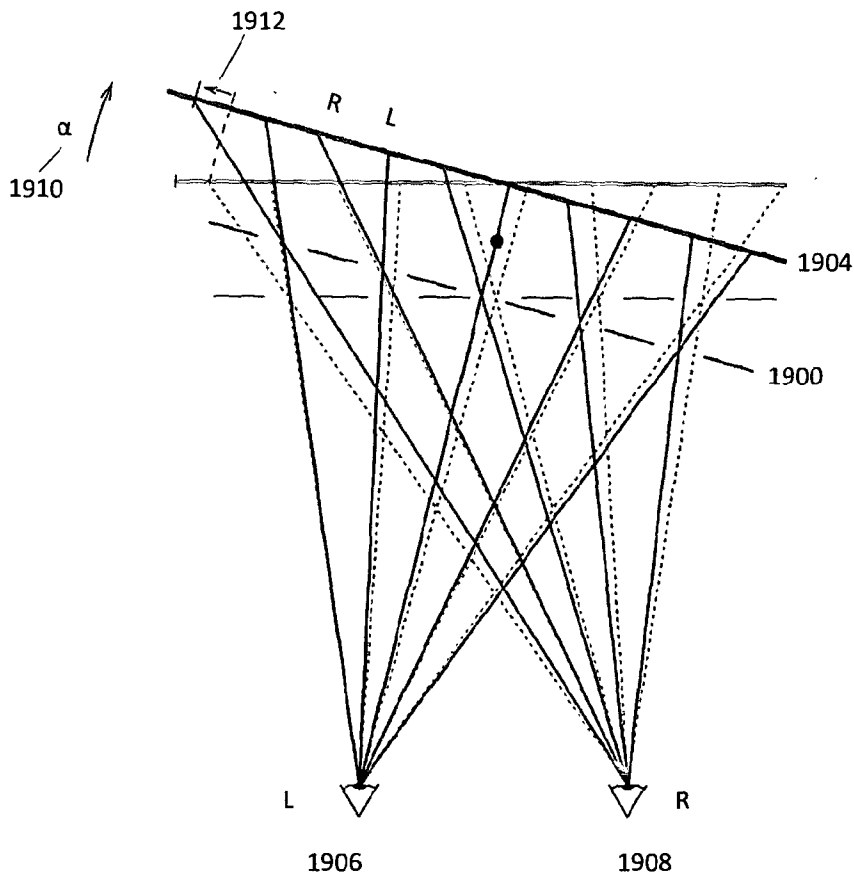
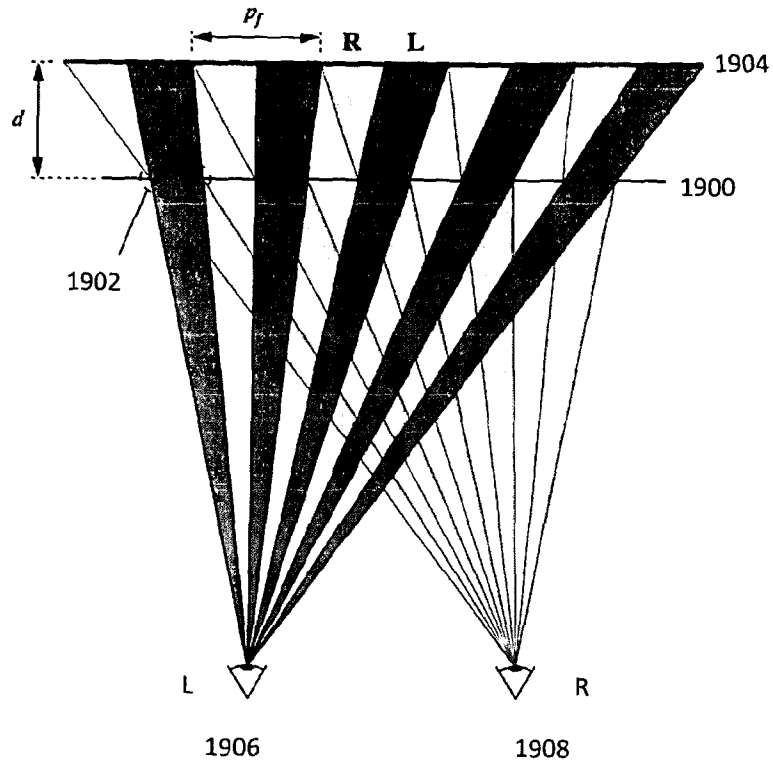


Figure 19

14 / 16

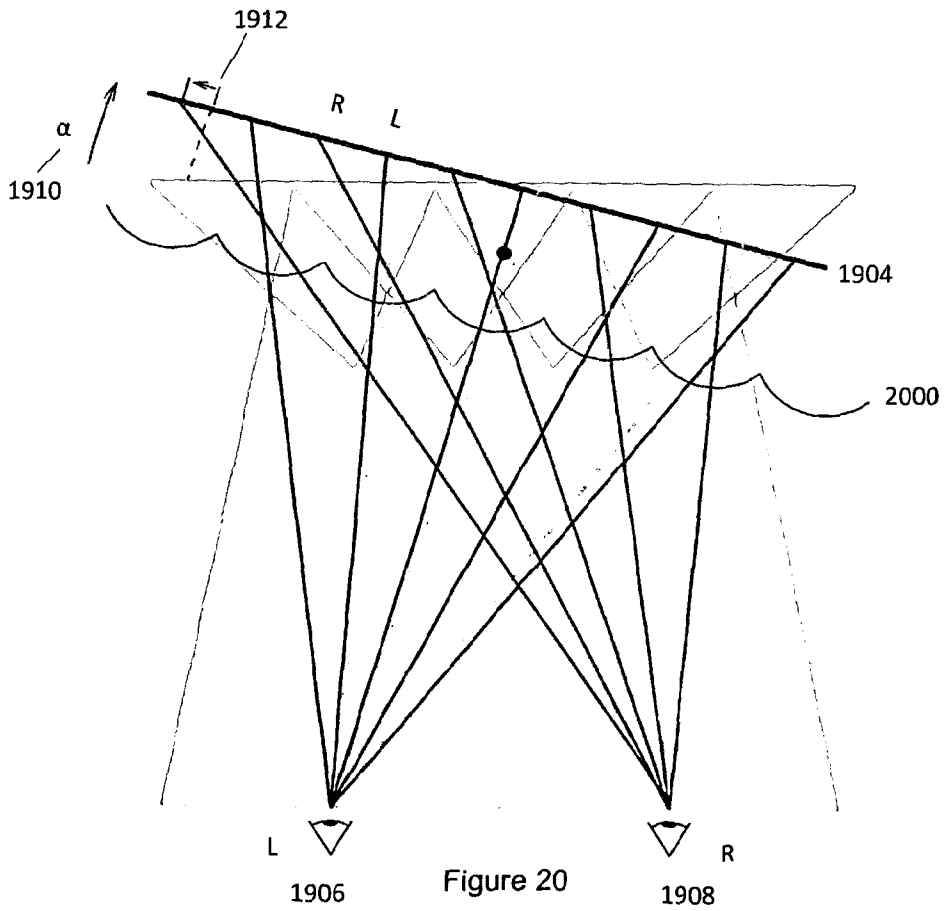
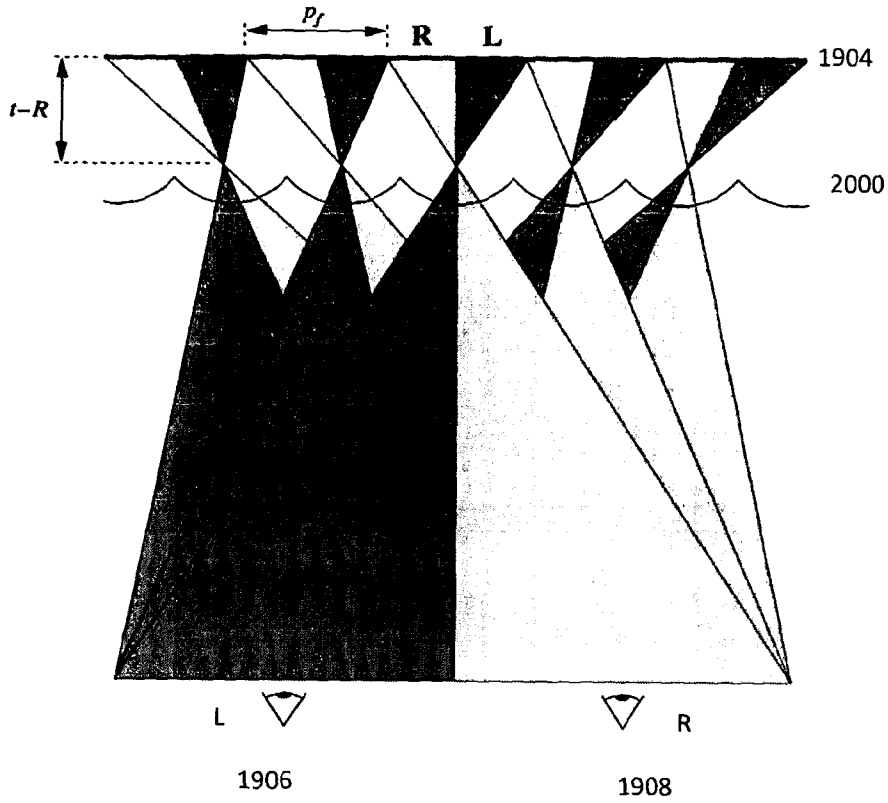


Figure 20

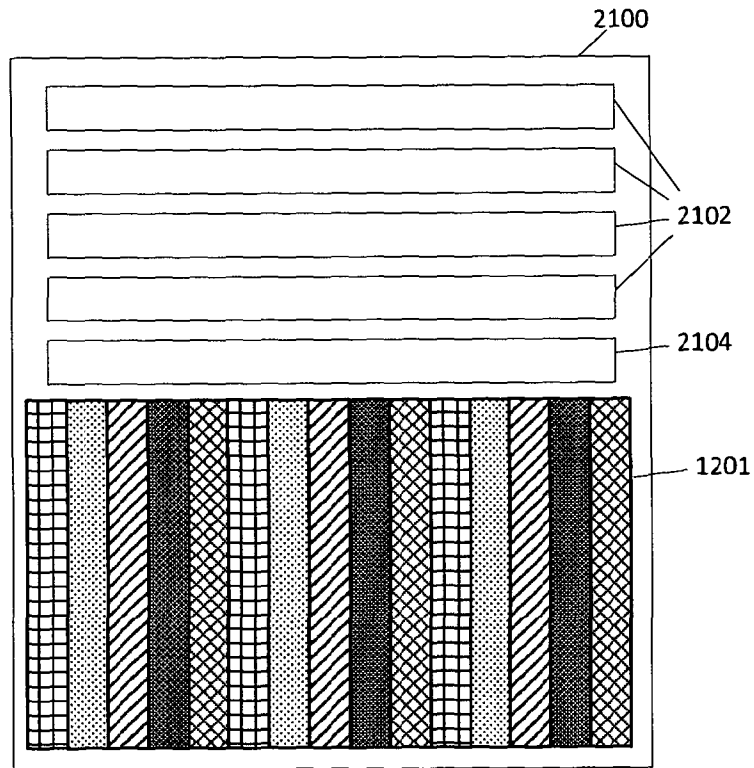


Figure 21

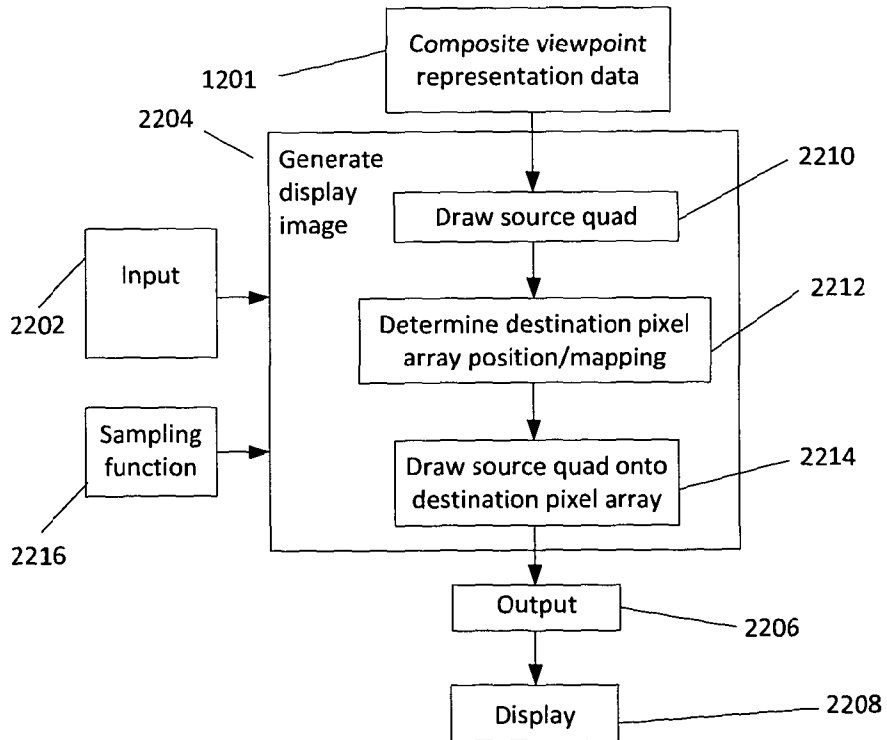


Figure 22

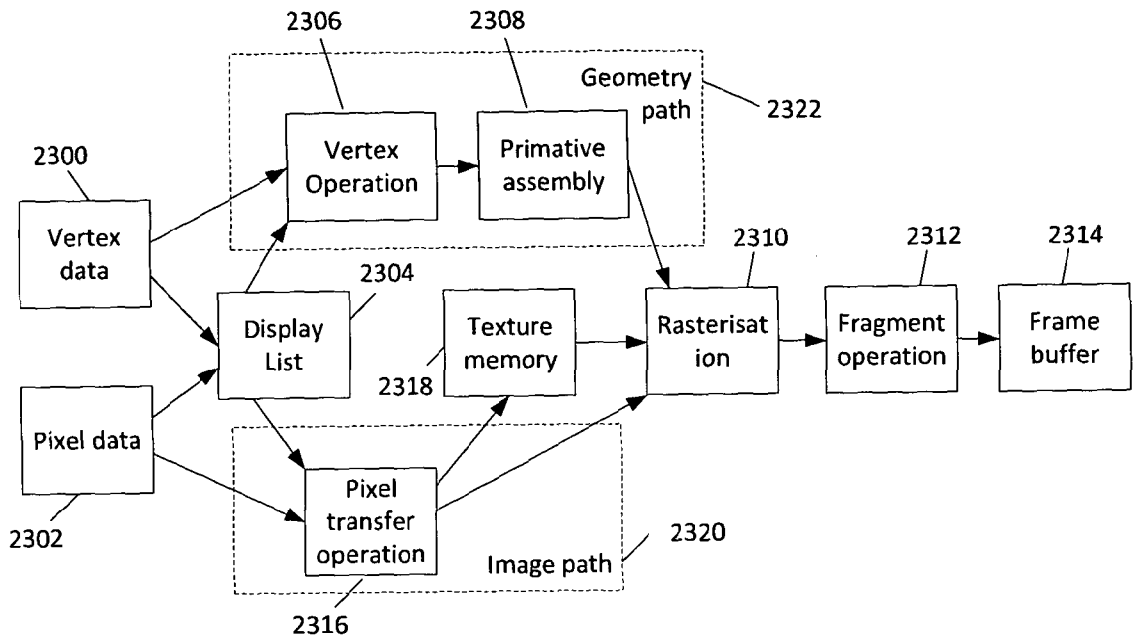


Figure 23

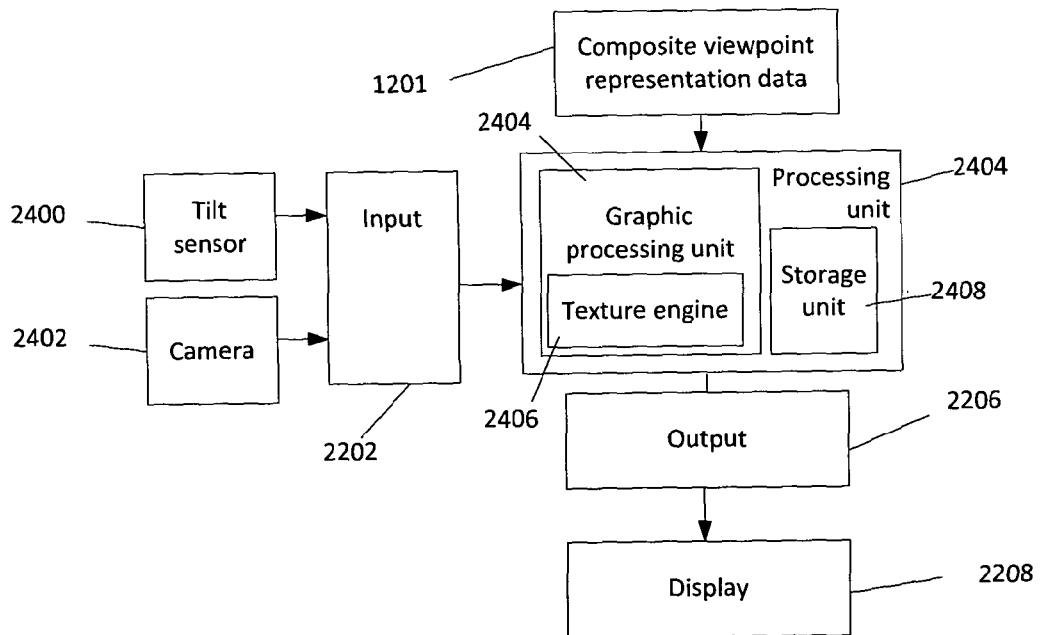


Figure 24