



(12) **CORRECTED EUROPEAN PATENT SPECIFICATION**

(15) Correction information:
Corrected version no 1 (W1 B1)
Corrections, see
Description Paragraph(s) 28

(51) Int Cl.:
G05B 19/4069 ^(2006.01) **B23Q 15/00** ^(2006.01)
G06T 15/00 ^(2011.01)

(48) Corrigendum issued on:
11.12.2019 Bulletin 2019/50

(86) International application number:
PCT/JP2009/059547

(45) Date of publication and mention
of the grant of the patent:
21.08.2019 Bulletin 2019/34

(87) International publication number:
WO 2009/145155 (03.12.2009 Gazette 2009/49)

(21) Application number: **09754663.4**

(22) Date of filing: **25.05.2009**

(54) **CUTTING PROCESS SIMULATION DISPLAY DEVICE, METHOD FOR DISPLAYING CUTTING
PROCESS SIMULATION, AND CUTTING PROCESS SIMULATION DISPLAY PROGRAM**

ANZEIGEVORRICHTUNG ZUR SIMULATION EINES SCHNEIDVERFAHRENS, VERFAHREN ZUR
ANZEIGE DER SIMULATION EINES SCHNEIDVERFAHRENS UND PROGRAMM ZUR ANZEIGE
DER SIMULATION EINES SCHNEIDVERFAHRENS

DISPOSITIF D'AFFICHAGE DE SIMULATION DE PROCESSUS DE DÉCOUPE, PROCÉDÉ POUR
AFFICHER UNE SIMULATION DE PROCESSUS DE DÉCOUPE, ET PROGRAMME D'AFFICHAGE
DE SIMULATION DE PROCESSUS DE DÉCOUPE

(84) Designated Contracting States:
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR
HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL
PT RO SE SI SK TR**

(56) References cited:
JP-A- 2 236 676 JP-A- 2008 083 973
US-A1- 2002 133 264

(30) Priority: **29.05.2008 JP 2008141471**

(43) Date of publication of application:
09.02.2011 Bulletin 2011/06

(73) Proprietor: **Mitsubishi Electric Corporation**
Chiyoda-ku
Tokyo 100-8310 (JP)

(72) Inventor: **MIYATA, Akira**
Tokyo 100-8310 (JP)

(74) Representative: **Zech, Stefan Markus**
Meissner Bolte Patentanwälte
Rechtsanwälte Partnerschaft mbB
Postfach 86 06 24
81633 München (DE)

- **S. TAKATA ET AL: "A Cutting Simulation System
for Machinability Evaluation Using a Workpiece
Model", CIRP ANNALS - MANUFACTURING
TECHNOLOGY, vol. 38, no. 1, 1989, pages
417-420, XP055185603, ISSN: 0007-8506, DOI:
10.1016/S0007-8506(07)62736-X**
- **PARKER S ET AL: "INTERACTIVE RAY TRACING
FOR VOLUME VISUALIZATION", IEEE
TRANSACTIONS ON VISUALIZATION AND
COMPUTER GRAPHICS, IEEE SERVICE
CENTER, LOS ALAMITOS, CA, US, vol. 5, no. 3,
July 1999 (1999-07), pages 238-250,
XP000865306, ISSN: 1077-2626, DOI:
10.1109/2945.795215**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

- RONI YAGEL ET AL: "Accelerating volume animation by space-leaping", VISUALIZATION '94; [ANNUAL IEEE CONFERENCE ON VISUALIZATION], IEEE COMPUTER SOCIETY PRESS, 10662 LOS VAQUEROS CIRCLE PO BOX 3014 LOS ALAMITOS, CA 90720-1264 USA, 25 October 1993 (1993-10-25), pages 62-69, XP058275028, ISSN: 1070-2385 ISBN: 978-0-7803-7200-9

Description**TECHNICAL FIELD**

[0001] The present invention relates to a device for displaying a cutting simulation, a method for displaying a cutting simulation, and a program for displaying a cutting simulation. In the cutting simulation, consecutive changes in the shape of a workpiece being cut are simulated on a computer and displayed on the screen of the display device in conjunction with the shape of a tool being used at that time.

BACKGROUND ART

[0002] US 2002/133264 A1 relates to a system for creating a design model, and machine instructions for building the same, using a virtual-reality-environment user interface. RONI YAGEL ET AL.: "Accelerating volume animation by space-leaping", the Ohio State University Department of Computer and Information Science, 25 October 1993, describes a method for speeding the process of volume animation. In cutting a workpiece on a machine tool such as a machining center, a cutting simulation performed to simulate changes in the shape of the workpiece on a computer is required to efficiently process the changes in the shape of the workpiece and display consecutive changes in the shape of the workpiece at high speeds on a screen. In general, the cut surface created by cutting operations may have a complicated shape that is defined by the relative motion between the cutting tool and the workpiece. Therefore, the workpiece shape is often represented, for example, in a voxel model or a dixel model that employs a set of so-called cells or minute cubes or prisms.

[0003] By way of example, a method for performing high-speed simulations using the voxel or dixel model is disclosed in Patent Document 1. The method uses the shape data that is obtained by converting the three-dimensional workpiece model into a dixel model. Then, the method calculates a sweep shape formed when the tool moves along a travel path and calculates the wedged and cylindrical shapes that constitute the sweep shape. Then, the respective bottom surfaces are turned into a polygon to remove a hidden surface using three-dimensional graphics hardware capable of depth buffering, thereby creating an intermediate image. At this time, the upper edge of a dixel is cut off on the basis of a depth value stored in the depth buffer. The upper end portion of the dixel is then turned into a polygon, whose hidden surface is in turn eliminated using the three-dimensional graphics hardware, thereby creating a final image for display on a screen.

PRIOR ART DOCUMENTS**PATENT DOCUMENTS**

- 5 **[0004]** Patent Document 1: Japanese Patent No. 3571564

DISCLOSURE OF INVENTION10 **PROBLEM TO BE SOLVED BY THE INVENTION**

[0005] In general, such a method as found in the conventional technology mentioned above is referred to as a polygon rendering scheme in which a displayed object is turned into a polygon to obtain an image with its hidden surface removed with the help of depth buffering. The polygon rendering scheme will take a long time to render polygons that are equivalent in number to the cells forming the voxel or dixel model employed for shape representations, though use of these models allow data structures to be simplified and internal shape processing to be performed more effectively. Accordingly, such a polygon rendering scheme is often based on the use of three-dimensional graphics hardware as in the aforementioned conventional art. However, such high-end three-dimensional graphics hardware may not be used in many cases by comparatively inexpensive computers often found at the place of machining operations.

[0006] On the other hand, the polygon rendering scheme described above can be replaced with another rendering scheme, i.e., the ray-tracing method. As shown in FIG. 2-1, in the ray-tracing method, a ray 203 is projected from each pixel 202 on the projection plane 201 in the direction of line of sight. Then, at the intersection 205 of the ray 203 and the object surface, color information (pixel values such as RGB) is obtained by calculation, thereby providing a display image. To use the ray-tracing method for displaying a voxel model 204, cells 206 found on the way to the object surface have to be sequentially traced as shown in FIG. 2-2, thereby determining whether the ray 203 intersects the cells. The determinations of the presence of intersections can be made relatively simply, but result in a considerable amount of computation as a whole because a large number of cells exist and the ray-tracing is repeated for each and every pixel.

[0007] As such, the ray-tracing method can be employed to simulate and display on a screen the shape of a workpiece, which is represented in the voxel model during its cutting operation. In this case, a considerable amount of computation required for ray tracing makes it difficult to perform these operations on an inexpensive, low-performance computer that has no three-dimensional graphics hardware.

[0008] The present invention was developed in view of the aforementioned problems. It is an object of the present invention to provide a device for displaying a cutting simulation, a method for displaying a cutting simulation, and a program for displaying a cutting simulation,

which can reduce the amount of computation required for ray-tracing and allow changes in the shape of a workpiece to be easily simulated even on an inexpensive, low-performance computer with no three-dimensional graphics hardware.

MEANS FOR SOLVING PROBLEM

[0009] In order to solve above-mentioned problems and to achieve the object, a device for displaying a cutting simulation according to claim 1 is proposed.

[0010] The rendered workpiece image update section according to the present invention starts ray tracing at a location defined by a non-updated previous depth distance and coordinates of a ray-traced pixel. Further aspects of the invention are defined in the claims 2 and 3.

[0011] According to the invention, the refreshing of a region, in which the tool was previously rendered, and the display updating of the workpiece with an actual change in the shape are separated from the display updating of a portion in which the tool is currently rendered. This minimizes the region that requires image rendering by ray-tracing. Furthermore, attention is also focused on the fact that a new surface created by cutting is always located at a deeper position than the previously viewed surface before cutting. This fact is reflected on the use of ray tracing to update a rendered image buffer portion and a depth buffer portion, which correspond to a rendered region associated with a change in the shape of the workpiece. Accordingly, the ray tracing is started from the location that is determined by the previous non-updated depth distance and the coordinates of a ray-traced pixel.

EFFECT OF THE INVENTION

[0012] According to the invention, the step of creating an image to be displayed on a screen is divided into two sub-steps of ray-tracing of a workpiece and ray-tracing of a tool. This allows for ray tracing of only the region that has to be truly updated, thereby reducing the amount of computation. Furthermore, the number of times of ray tracing of pixels which have to be updated from the previously displayed status is reduced, resulting in the total amount of computation being decreased. This allows cutting simulations to be performed even on an inexpensive low-performance computer.

[0013] Furthermore, in ray-tracing of pixels to be updated, the ray tracing starts at a location that is defined by a non-updated previous depth distance and coordinates of a ray-traced pixel. This allows for eliminating an unnecessary determination of the presence of an intersection between cell and ray. Thus, the number of determinations of cell intersections is reduced and the amount of computation per tracing for one update is decreased. It is thus possible to perform a cutting simulation even on an inexpensive low-performance computer.

BRIEF DESCRIPTION OF DRAWINGS

[0014]

5 FIG. 1 is a block diagram illustrating the entire configuration of an embodiment of a device for displaying a cutting simulation according to the present invention.

10 FIG. 2-1 is an explanatory view showing a method for displaying a voxel model by ray tracing, in which a ray is projected in the direction of line of sight from each pixel on a projection plane to calculate color information at an intersection of the ray and the object surface and thereby obtains a display image.

15 FIG. 2-2 is an explanatory view showing a method for displaying a voxel model by ray tracing, in which cells found on the way until the ray reaches an object surface are traced sequentially to determine whether there is an intersection of the ray and the cells.

20 FIG. 3 is a view illustrating an example of a screen display update in an embodiment of a device for displaying a cutting simulation according to the present invention.

25 FIG. 4-1 is an explanatory view illustrating the basic principle of ray tracing in a rendered workpiece image update section, showing the result of ray tracing for a previous screen display update.

30 FIG. 4-2 is an explanatory view illustrating the basic principle of ray tracing in the rendered workpiece image update section, showing the ray tracing for a current screen display update.

FIG. 5 is a flowchart illustrating the flow of ray tracing in the rendered workpiece image update section.

35 FIG. 6 is a flowchart illustrating the flow of ray tracing in a rendered tool image creation section.

BEST MODES FOR CARRYING OUT THE INVENTION

40 **[0015]** Now, embodiments of a device for displaying a cutting simulation, a method for displaying a cutting simulation, and a program for displaying a cutting simulation according to the present invention will be described in detail below with reference to the accompanying drawings. Note that the invention is not limited to these embodiments.

Embodiments

50 **[0016]** FIG. 1 is a block diagram illustrating the entire configuration of an embodiment of a device for displaying a cutting simulation according to the present invention. In FIG. 1, the cutting simulation display device mainly includes a simulation core section 10 and a simulation display section 20. The simulation core section 10 represents the shape of a workpiece using a voxel model and simulates changes in the shape of the workpiece by a cutting operation. The simulation display section 20 displays on the screen the shapes of the workpiece that

consecutively change.

[0017] The simulation core section 10 includes workpiece shape data 11, tool path data 12, and a cutting simulation section 13. The workpiece shape data 11 is stored in the storage of the device and represented in a voxel model. Similarly, the tool path data 12 is stored in the storage of the device and includes information of tool shapes and tool travel paths. The cutting simulation section 13 updates the workpiece shape data 11 on the basis of the tool path data 12.

[0018] On the other hand, the simulation display section 20 includes projection parameters 21, a rendered workpiece image buffer 22, a rendered workpiece depth buffer 23, a screen display frame buffer 24, updated rendering region management data 25, a rendered workpiece image update section 26, tool rendering region management data 27, a rendered tool image creation section 28, and an image transfer section 29. The projection parameters 21 include, for example, the direction of line of sight and illumination conditions for rendering the workpiece and the tool. The rendered workpiece image buffer 22 holds only an image of the workpiece rendered by projection in the direction of line of sight. The rendered workpiece depth buffer 23 holds the depth distance of each pixel from the projection plane to the workpiece surface when the single workpiece is rendered by projection in the direction of line of sight. The screen display frame buffer 24 holds a rendered image of the workpiece and the tool to display it on the screen such as a CRT. The updated rendering region management data 25 manages the region information required to be updated in the rendered workpiece image buffer 22 and the rendered workpiece depth buffer 23. The rendered workpiece image update section 26 recalculates the data of the rendered workpiece image buffer 22 and the rendered workpiece depth buffer 23 by ray tracing from the workpiece shape data (voxel) 11. The tool rendering region management data 27 manages a region in which the tool was rendered in the previous display update. The rendered tool image creation section 28 creates a projected rendering image of and near the tool by ray tracing on the basis of the tool shape in the tool path data 12 and at the end of the travel path, i.e., the current position of the tool. The image transfer section 29 transfers to the screen display frame buffer 24 the image data of an updated region in the rendered workpiece image buffer 22 and the rendered tool image created by the rendered tool image creation section 28.

[0019] Here, the rendered workpiece image buffer 22, the rendered workpiece depth buffer 23, and the screen display frame buffer 24 have the same number of pixels (width by height). Furthermore, the updated rendering region management data 25 represents an updated region with the coordinates of two vertices on the diagonal line and is initialized so as to be displayed on the entire screen at the beginning of a simulation.

[0020] At the beginning of a simulation, the workpiece shape data 11 represents the shape of an unprocessed

workpiece. Only at the start of the simulation, the rendered workpiece image update section 26 calculates and initializes by ray tracing the data of both the rendered workpiece image buffer 22 and the rendered workpiece depth buffer 23 for the entire display region. The image transfer section 29 then transfers the data of the entire region of the rendered workpiece image buffer 22 to the screen display frame buffer 24. After the initial display is completed in this manner, the updated rendering region management data 25 is cleared to null.

[0021] Now, operations during the simulation will be described. FIG. 3 illustrates a typical example of updated screen display during a simulation. This figure illustrates non-updated display contents or previous display contents denoted by 301 and the display contents resulting from the current display update denoted by 304. 306 shown by short dashed lines in the figure is a rendering region corresponding to a change in shape of a workpiece 302. 307 denotes a region in which a not-yet-displaced tool 303 was previously rendered, whereas 308 denotes a region in which a displaced tool 305 will be rendered. In updating screen display, related data has to be updated so that at least these regions 306 to 307 display the latest contents.

[0022] Now, the operation of each section will be described. The cutting simulation section 13 reads tool shape information and one step of path information from the tool path data 12 to simulate a cutting operation associated with a tool movement, and updates the workpiece shape data 11 based on the result. The cutting is simulated, for example, by performing a difference Boolean operation on the workpiece shape data 11 to subtract a swept tool shape, obtained by sweeping the tool shape along the path of movement, from the workpiece shape.

[0023] Each time one step of cutting simulation is completed, the cutting simulation section 13 sends, to the rendered workpiece image update section 26, the three-dimensional region data which has been changed by the cutting. Here, the three-dimensional region data is assumed to be a three-dimensional inclusion box surrounding the changed portion of the workpiece shape. The rendered workpiece image update section 26 calculates a two-dimensional updated region, which is obtained by projecting the received three-dimensional region data with the projection parameters 21, and updates the updated rendering region management data 25.

[0024] Now, the rendered workpiece image update section 26 calculates the pixel data (RGB values) and the depth distance by ray tracing for the pixels included in the updated region using the updated rendering region management data 25. The rendered workpiece image update section 26 then updates the rendered workpiece image buffer 22 and the rendered workpiece depth buffer 23.

[0025] Here, the coordinates of the current pixel to be processed are assumed to be (u, v). FIGS. 4-1 and 4-2 each are an explanatory cross-sectional view illustrating

the basic principle of ray tracing in the rendered workpiece image update section 26 when the projection plane is viewed horizontally from the side. FIG. 4-1 illustrates the result of the ray tracing in the previous screen display update. In the figure, the depth distance (depth distance w) 403 from the target pixel (pixel (u, v)) 402 on the projection plane 401 to the surface of the workpiece shape 404 is calculated and stored in the rendered workpiece depth buffer 23.

[0026] FIG. 4-2 illustrates the ray tracing in the current screen display update. In the figure, the current ray tracing starts at a point (ray tracing start point P) 406 spaced apart from the target pixel 402 by the depth distance 403. The ray tracing start point 406 is located on the surface of the workpiece shape at the time of the previous screen display update. However, when the cutting has changed the workpiece shape 404 in FIG. 4-1 into the workpiece shape 405 in FIG. 4-2, the new surface of the shape always appears at a deeper position in the direction of line of sight (see a ray (ray L) 407 in FIG. 4-2). Therefore, the ray tracing started at the point 406 still provides a proper result of rendering.

[Ray tracing by the rendered workpiece image update section (Ray tracing of workpiece)]

[0027] FIG. 5 is a flowchart illustrating the operation of the ray tracing in the rendered workpiece image update section 26. Note that symbols in the flowchart are used in a basic model for ray tracing (see FIGS. 4-1 and 4-2). Referring to FIG. 5, in step S101, the rendered workpiece image update section 26 first reads the rendered workpiece depth buffer 23 for the depth distance w to the pixel of interest. Next, in step S102, a three-dimensional coordinate point q(u, v, w) obtained by combining the pixel coordinates (u, v) and the depth distance w is converted into the coordinate system of the workpiece shape data to find the three-dimensional coordinates P(x, y, z). Subsequently, in step S103, a line segment L perpendicular to the projection plane and starting at a point P in the direction of depth is calculated as ray data. In step S104, the voxel model as workpiece shape data is searched for cell C which includes the point P.

[0028] Ray tracing takes place in the loop from steps S105 to S107. In step S105, it is determined whether the ray L intersects the surface of the workpiece shape within the cell C of interest. If "Yes," the process exits the loop and proceeds to step S108. If "No," it is determined in step S106 whether there is a cell adjacent to the cell C of interest via an interface, on which the cell C intersects the ray L in a deeper position among 6 interfaces of the cell C. If there is an adjacent cell, the adjacent cell is employed as a new cell C of interest in step S107, and the process returns to step S105. If there is no adjacent cell, the process moves to step S111.

[0029] Steps S108 to S110 are applied to a case where the ray L intersects the surface of the workpiece shape within the cell C of interest. In step S108, the point of

intersection X is calculated. Next, in step S109, the pixel data (the luminance of each of RGB) is calculated based on the normal vector to the workpiece shape and the illumination condition at the point of intersection X from the projection parameters 21, and then stored at the associated position in the rendered workpiece image buffer 22. In step S110, the distance from the projection plane to the point of intersection X (the depth distance) is calculated and stored at the associated position in the rendered workpiece depth buffer 23. Then, the process exits the ray tracing.

[0030] Step S111 is applied to the case where the ray L passes through the voxel model without intersecting the workpiece shape. In this case, a predetermined background color is stored at the associated position in the rendered workpiece image buffer 22, and a background distance of a sufficiently large value is stored at the associated position in the rendered workpiece depth buffer 23.

[0031] Note that the present embodiment is configured such that the rendered workpiece image buffer 22 and the rendered workpiece depth buffer 23 hold the rendered image data and the depth distance data of the workpiece shape to be cut. However, it is also acceptable to store data created by combining the projected rendered image and the depth distance data calculated relative to another object, such as a workpiece fixture, which is located at an invariant position relative to the workpiece during a cutting simulation.

[0032] The rendered tool image creation section 28 creates a projected rendered image of and near the tool. This is done when the data associated with all the pixels to be updated in the rendered workpiece image buffer 22 and the rendered workpiece depth buffer 23 has been updated by ray tracing as described above. The current processing also creates images by ray tracing.

[0033] The target region in which a rendered tool image is produced is a two-dimensional region on which the three-dimensional inclusion box surrounding the tool shape at the current position is projected with the projection parameters 21 on the basis of the tool shape information and the current tool position (the end of a travel path) stored in the tool path data 12. For the pixels included in this two-dimensional region, an image is created by ray tracing.

[Ray tracing by the rendered tool image creation section (ray tracing for tool)]

[0034] FIG. 6 is a flowchart illustrating the operation of ray tracing in the rendered tool image creation section 28. Note that symbols in the flowchart are used in a basic model of ray tracing (see FIGS. 4-1 and 4-2). Referring to FIG. 6, in step S201, the rendered tool image creation section 28 first calculates, as ray data, a line segment L perpendicular to the projection plane and starting from the pixel to be processed in the direction of depth. Next, in step S202, it is determined whether the ray L intersects

the surface of the tool shape. If "Yes," the process proceeds to step S203. In step S203, the point of intersection X of the ray L and the surface of the tool shape is calculated. Next, in step S204, the depth distance d from the projection plane to the point of intersection X is calculated. Then, in step S205, the depth distance w associated with the pixel of interest is read from the rendered workpiece depth buffer 23. In step S206, the depth distances d and w are compared.

[0035] Step S207 is applied to the case where $d < w$ in step S206, i.e., the tool shape is positioned before the workpiece shape. In this case, the pixel data (the luminance of each of RGB) is calculated based on the normal vector to the tool shape and the illumination condition at the point of intersection X from the projection parameters 21. Then, the pixel data is stored as pixel data of a rendered tool image. Step S208 is applied to the case where the ray L doesn't intersect the surface of the tool shape in step S202 or where $d \geq w$ in step S206, i.e., the tool shape is located in a deeper position than the workpiece shape. In this case, the pixel data associated with the pixel of interest is read from the rendered workpiece image buffer 22 and employed as the pixel data of the rendered tool image.

[0036] Finally, the image transfer section 29 transfers, to the screen display frame buffer 24, partial image data in the rendered workpiece image buffer 22 and the rendered tool image data created in the rendered tool image creation section 28.

[0037] Here, the partial image data of the regions corresponding to the updated region stored in the updated rendering region management data 25 and the previous tool rendering region stored in the tool rendering region management data 27 is read from the rendered workpiece image buffer 22, and then transferred to the screen display frame buffer 24. Subsequently, the rendered tool image data created by the rendered tool image creation section 28 is transferred to the screen display frame buffer 24. When the image transfer to the screen display frame buffer 24 is completed, the image transfer section 29 clears the updated rendering region management data 25 and sets the region of the current rendered tool image to the tool rendering region management data 27.

[0038] As described above, according to the present embodiment, the image to be displayed on the screen is created in two processing steps: workpiece ray tracing and tool ray tracing. This allows for performing the ray tracing only on the region truly required to be updated, thereby reducing the amount of computation. Furthermore, as described above, the ray tracing of the workpiece and the tool can be performed separately in this manner, thereby making each ray tracing adequate and efficient. Furthermore, the workpiece ray tracing uses the previously determined depth distance to eliminate the need of an unnecessary determination of an intersection of the ray and the cell, thereby reducing the amount of computation.

[0039] In the present embodiment, the device for dis-

playing a cutting simulation is configured to include the rendered workpiece image update section 26 for ray tracing of the workpiece and the rendered tool image creation section 28 for ray tracing of the tool. However, it is also acceptable to update only rendered images of a change in the workpiece shape without the rendered tool image creation section 28. This arrangement eliminates the need of creating a rendered tool image and transferring an image of a tool rendering region, thus further reducing the amount of computation.

[0040] As such, a cutting simulation system according to the present embodiment allows a cutting simulation by representing the shape of a workpiece in the voxel model and simulating changes in the workpiece shape made by cutting operations. This cutting simulation can be performed even with a small amount of computation, thus implementing the cutting simulation even on an inexpensive low-performance computer.

20 INDUSTRIAL APPLICABILITY

[0041] As described above, the present invention can be effectively applied to a simulation which employs a voxel model to represent the shape of an object and simulate it on a computer. In particular, the invention is suitable to a cutting simulation in which a computer simulates changes in the workpiece shape or a complicated cut surface that is created during a cutting operation and defined by the relative motion between the tool and the workpiece.

EXPLANATIONS OF LETTERS OR NUMERALS

[0042]

- 10 Simulation core section
- 11 Workpiece shape data represented in voxel model
- 12 Tool path data including information of tool shapes and tool travel paths
- 13 Cutting simulation section
- 20 Simulation display section
- 21 Projection parameters
- 22 Rendered workpiece image buffer
- 23 Rendered workpiece depth buffer
- 24 Screen display frame buffer
- 25 Updated rendering region management data
- 26 Rendered workpiece image update section
- 27 Tool rendering region management data
- 28 Rendered tool image creation section
- 29 Image transfer section
- 201 Projection plane
- 202 Target pixel for ray tracing
- 203 Ray
- 204 Voxel model
- 205 Intersection of ray and object
- 206 Cell found on a way of ray
- 301 Previous rendering

302 Workpiece
 303 Not-yet-displaced tool
 304 Current rendering
 305 Displaced tool
 306 Region in which the workpiece shape was changed 5
 307 Region in which the tool was previously rendered
 308 Region in which the tool will be rendered
 401 Projection plane 10
 402 Pixel
 403 Depth distance
 404, 405 Workpiece shape
 406 Ray-tracing start point
 407 Ray 15

Claims

1. A device for displaying a cutting simulation for representing a shape of a workpiece in a voxel model to simulate a change in the shape of the workpiece by a cutting operation, the device comprising:
 - a rendered workpiece image update section (26) for managing a rendered workpiece image buffer (22) and a rendered workpiece depth buffer (23) to hold image data and depth distance data of a projected image of the workpiece and updating by ray tracing a portion in the rendered workpiece image buffer (22) and the rendered workpiece depth buffer (23), the portion being associated with a rendering region corresponding to a change in the shape of the workpiece, wherein the rendered workpiece image update section (26) starts ray tracing at a location on the surface of the workpiece shape at the time of the previous screen display update defined by a depth distance (w) and coordinates (u,v) of a ray-traced pixel, wherein the rendered workpiece image update section 26 first reads the rendered workpiece depth buffer 23 for the depth distance (w) to the ray-traced pixel;
 - a three-dimensional coordinate point $q(u, v, w)$ obtained by combining the pixel coordinates (u, v) and the depth distance (w) is converted into the coordinate system of the workpiece shape data to find the three-dimensional coordinate point $P(x, y, z)$, and
 - a line segment (L) perpendicular to the projection plane associated with the projected image and starting at the point P in the direction of depth is calculated as ray data.
2. A method for displaying a cutting simulation for representing a shape of a workpiece in a voxel model to simulate a change in the shape of the workpiece

by a cutting operation, the method comprising:

a rendered workpiece image update step of managing a rendered workpiece image buffer (22) and a rendered workpiece depth buffer (23) to hold image data and depth distance data of a projected image of the workpiece and updating by ray tracing a portion in the rendered workpiece image buffer (22) and the rendered workpiece depth buffer (23), the portion being associated with a rendering region corresponding to a change in the shape of the workpiece, wherein the rendered workpiece image update step includes starting ray tracing at a location on the surface of the workpiece shape at the time of the previous screen display update defined by a depth distance (w) and coordinates (u, v) of a ray-traced pixel, wherein the rendered workpiece image update section (26) first reads the rendered workpiece depth buffer (23) for the depth distance (w) to the ray-traced pixel; a three-dimensional coordinate point $q(u, v, w)$ obtained by combining the pixel coordinates (u, v) and the depth distance (w) is converted into the coordinate system of the workpiece shape data to find the three-dimensional coordinate point $P(x, y, z)$, and a line segment (L) perpendicular to the projection plane associated with the projected image and starting at the point P in the direction of depth is calculated as ray data.

3. A program executed on a computer for displaying a cutting simulation for representing a shape of a workpiece in a voxel model to simulate a change in the shape of the workpiece by a cutting operation, the program comprising:

a rendered workpiece image update procedure of managing a rendered workpiece image buffer and a rendered workpiece depth buffer to hold image data and depth distance data of a projected image of the workpiece and updating by ray tracing a portion in the rendered workpiece image buffer and the rendered workpiece depth buffer, the portion being associated with a rendering region corresponding to a change in the shape of the workpiece, wherein the rendered workpiece image update procedure includes starting ray tracing at a location on the surface of the workpiece shape at the time of the previous screen display update defined by a depth distance (w) and coordinates (u, v) of a ray-traced pixel, wherein the rendered workpiece image update section 26 first reads the rendered workpiece depth buffer 23 for the depth distance (w) to the

ray-traced pixel;
 a three-dimensional coordinate point $q(u, v, w)$ obtained by combining the pixel coordinates (u, v) and the depth distance (w) is converted into the coordinate system of the workpiece shape data to find the three-dimensional coordinate point $P(x, y, z)$, and
 a line segment (L) perpendicular to the projection plane associated with the projected image and starting at the point P in the direction of depth is calculated as ray data.

Patentansprüche

1. Vorrichtung zum Anzeigen einer Schneidesimulation zum Darstellen einer Form eines Werkstücks in einem Voxel-Modell, um eine Änderung an der Form des Werkstücks durch einen Schneidevorgang zu simulieren, wobei die Vorrichtung umfasst:
 einen Wiedergabewerkstückbildaktualisierungsabschnitt (26) zum Verwalten eines Wiedergabewerkstückbildspeichers (22) und eines Wiedergabewerkstücktiefspeichers (23), um Bilddaten und Tiefendistanzdaten eines projizierten Bilds des Werkstücks vorzuhalten und durch Strahlverfolgung (*ray tracing*) einen Abschnitt in dem Wiedergabewerkstückbildspeicher (22) und dem Wiedergabewerkstücktiefspeicher (23) zu aktualisieren, wobei der Abschnitt mit einem Wiedergabebereich zusammenhängt, der einer Änderung an der Form des Werkstücks entspricht, wobei der Wiedergabewerkstückbildaktualisierungsabschnitt (26) die Strahlverfolgung an einer Stelle auf der Oberfläche der Werkstückform zum Zeitpunkt der vorherigen Bildschirmaktualisierung beginnt, die durch eine Tiefendistanz (w) und Koordinaten (u, v) eines strahlverfolgten Pixels definiert ist, wobei der Wiedergabewerkstückbildaktualisierungsabschnitt (26) zuerst den Wiedergabewerkstücktiefspeicher (23) auf die Tiefendistanz (w) zum strahlverfolgten Pixel hin ausliest; ein dreidimensionaler Koordinatenpunkt $q(u, v, w)$, der erhalten wird, indem die Pixel-Koordinaten (u, v) und die Tiefendistanz (w) kombiniert werden, in das Koordinatensystem der Werkstückformdaten umgesetzt wird, um den dreidimensionalen Koordinatenpunkt $P(x, y, z)$ zu finden, und ein Liniensegment (L) senkrecht zu der Projektionsebene, die mit dem projizierten Bild zusammenhängt und an dem Punkt P in der Tiefenrichtung beginnt, als Strahl Daten berechnet wird.

2. Verfahren zum Anzeigen einer Schneidesimulation zum Darstellen einer Form eines Werkstücks in einem Voxel-Modell, um eine Änderung an der Form des Werkstücks durch einen Schneidevorgang zu simulieren, wobei das Verfahren umfasst:

einen Wiedergabewerkstückbildaktualisierungsschritt, bei dem ein Wiedergabewerkstückbildspeicher (22) und ein Wiedergabewerkstücktiefspeicher (23) verwaltet werden, um Bilddaten und Tiefendistanzdaten eines projizierten Bilds des Werkstücks vorzuhalten und durch Strahlverfolgung einen Abschnitt in dem Wiedergabewerkstückbildspeicher (22) und dem Wiedergabewerkstücktiefspeicher (23) zu aktualisieren, wobei der Abschnitt mit einem Wiedergabebereich zusammenhängt, der einer Änderung an der Form des Werkstücks entspricht, wobei der Wiedergabewerkstückbildaktualisierungsschritt umfasst, die Strahlverfolgung an einer Stelle auf der Oberfläche der Werkstückform zum Zeitpunkt der vorherigen Bildschirmaktualisierung zu beginnen, die durch eine Tiefendistanz (w) und Koordinaten (u, v) eines strahlverfolgten Pixels definiert ist, wobei der Wiedergabewerkstückbildaktualisierungsabschnitt (26) zuerst den Wiedergabewerkstücktiefspeicher (23) auf die Tiefendistanz (w) zum strahlverfolgten Pixel hin ausliest; ein dreidimensionaler Koordinatenpunkt $q(u, v, w)$, der erhalten wird, indem die Pixel-Koordinaten (u, v) und die Tiefendistanz (w) kombiniert werden, in das Koordinatensystem der Werkstückformdaten umgesetzt wird, um den dreidimensionalen Koordinatenpunkt $P(x, y, z)$ zu finden, und ein Liniensegment (L) senkrecht zu der Projektionsebene, die mit dem projizierten Bild zusammenhängt und an dem Punkt P in der Tiefenrichtung beginnt, als Strahl Daten berechnet wird.

3. Programm, das auf einem Computer zum Anzeigen einer Schneidesimulation zum Darstellen einer Form eines Werkstücks in einem Voxel-Modell abläuft, um eine Änderung an der Form des Werkstücks durch einen Schneidevorgang zu simulieren, wobei das Programm umfasst:

eine Wiedergabewerkstückbildaktualisierungsprozedur, bei der ein Wiedergabewerkstückbildspeicher und ein Wiedergabewerkstücktiefspeicher verwaltet werden, um Bilddaten und Tiefendistanzdaten eines projizierten Bilds des Werkstücks vorzuhalten und durch Strahlverfolgung einen Abschnitt in dem Wiedergabewerkstückbildspeicher und dem Wiedergabewerk-

stücktiefenspeicher zu aktualisieren, wobei der Abschnitt mit einem Wiedergabebereich zusammenhängt, der einer Änderung an der Form des Werkstücks entspricht, wobei die Wiedergabewerkstückbildaktualisierungsprozedur umfasst, die Strahlverfolgung an einer Stelle auf der Oberfläche der Werkstückform zum Zeitpunkt der vorherigen Bildschirmaktualisierung zu beginnen, die durch eine Tiefendistanz (w) und Koordinaten (u, v) eines strahlverfolgten Pixels definiert ist, wobei der Wiedergabewerkstückbildaktualisierungsabschnitt (26) zuerst den Wiedergabewerkstücktiefenspeicher (23) auf die Tiefendistanz (w) zum strahlverfolgten Pixel hin ausliest; ein dreidimensionaler Koordinatenpunkt q (u, v, w), der erhalten wird, indem die Pixel-Koordinaten (u, v) und die Tiefendistanz (w) kombiniert werden, in das Koordinatensystem der Werkstückformdaten umgesetzt wird, um den dreidimensionalen Koordinatenpunkt P (x, y, z) zu finden, und ein Liniensegment (L) senkrecht zu der Projektionsebene, die mit dem projizierten Bild zusammenhängt und an dem Punkt P in der Tiefenrichtung beginnt, als Strahl Daten berechnet wird.

Revendications

1. Dispositif d'affichage d'une simulation de coupe destinée à représenter une forme d'une pièce à usiner dans un modèle voxel pour simuler un changement dans la forme de la pièce à usiner par une opération de coupe, le dispositif comprenant :

une section de mise à jour d'image rendue de pièce à usiner (26) destinée à gérer un tampon d'image rendue de pièce à usiner (22) et un tampon de profondeur rendue de pièce à usiner (23) pour garder des données d'image et des données de distance de profondeur d'une image projetée de la pièce à usiner et à mettre à jour, par lancer de rayon, une partie dans le tampon d'image rendue de pièce à usiner (22) et le tampon de profondeur rendue de pièce à usiner (23), la partie étant associée à une région de rendu correspondant à un changement dans la forme de la pièce à usiner, sachant que la section de mise à jour d'image rendue de pièce à usiner (26) démarre un lancer de rayon à un emplacement sur la surface de la forme de pièce à usiner au moment de la précédente mise à jour d'affichage d'écran défini par une distance de profondeur (w) et des coordonnées (u, v) d'un pixel obtenu par lancer de rayon,

sachant que la section de mise à jour d'image rendue de pièce à usiner (26) lit d'abord le tampon de profondeur rendue de pièce à usiner (23) pour la distance de profondeur (w) au pixel obtenu par lancer de rayon ; un point de coordonnées tridimensionnel q (u, v, w) obtenu en combinant les coordonnées de pixel (u, v) et la distance de profondeur (w) est converti dans le système de coordonnées des données de forme de pièce à usiner pour trouver le point de coordonnées tridimensionnel P (x, y, z), et un segment de ligne (L) perpendiculaire au plan de projection associé à l'image projetée et commençant au point P dans la direction de profondeur est calculé comme données de rayon.

2. Procédé d'affichage d'une simulation de coupe destinée à représenter une forme d'une pièce à usiner dans un modèle voxel pour simuler un changement dans la forme de la pièce à usiner par une opération de coupe, le procédé comprenant :

une étape de mise à jour d'image rendue de pièce à usiner consistant à gérer un tampon d'image rendue de pièce à usiner (22) et un tampon de profondeur rendue de pièce à usiner (23) pour garder des données d'image et des données de distance de profondeur d'une image projetée de la pièce à usiner et à mettre à jour, par lancer de rayon, une partie dans le tampon d'image rendue de pièce à usiner (22) et le tampon de profondeur rendue de pièce à usiner (23), la partie étant associée à une région de rendu correspondant à un changement dans la forme de la pièce à usiner, sachant que l'étape de mise à jour d'image rendue de pièce à usiner inclut le démarrage d'un lancer de rayon à un emplacement sur la surface de la forme de pièce à usiner au moment de la précédente mise à jour d'affichage d'écran défini par une distance de profondeur (w) et des coordonnées (u, v) d'un pixel obtenu par lancer de rayon, sachant que la section de mise à jour d'image rendue de pièce à usiner (26) lit d'abord le tampon de profondeur rendue de pièce à usiner (23) pour la distance de profondeur (w) au pixel obtenu par lancer de rayon ; un point de coordonnées tridimensionnel q (u, v, w) obtenu en combinant les coordonnées de pixel (u, v) et la distance de profondeur (w) est converti dans le système de coordonnées des données de forme de pièce à usiner pour trouver le point de coordonnées tridimensionnel P (x, y, z), et un segment de ligne (L) perpendiculaire au plan de projection associé à l'image projetée et com-

mençant au point P dans la direction de profondeur est calculé comme données de rayon.

3. Programme exécuté sur un ordinateur pour afficher une simulation de coupe destinée à représenter une forme d'une pièce à usiner dans un modèle voxel pour simuler un changement dans la forme de la pièce à usiner par une opération de coupe, le programme comprenant :
- une procédure de mise à jour d'image rendue de pièce à usiner consistant à gérer un tampon d'image rendue de pièce à usiner et un tampon de profondeur rendue de pièce à usiner pour garder des données d'image et des données de distance de profondeur d'une image projetée de la pièce à usiner et à mettre à jour, par lancer de rayon, une partie dans le tampon d'image rendue de pièce à usiner et le tampon de profondeur rendue de pièce à usiner, la partie étant associée à une région de rendu correspondant à un changement dans la forme de la pièce à usiner,
- sachant que la procédure de mise à jour d'image rendue de pièce à usiner inclut le démarrage du lancer de rayon à un emplacement sur la surface de la forme de pièce à usiner au moment de la précédente mise à jour d'affichage d'écran défini par une distance de profondeur (w) et des coordonnées (u, v) d'un pixel obtenu par lancer de rayon,
- sachant que la section de mise à jour d'image rendue de pièce à usiner (26) lit d'abord le tampon de profondeur rendue de pièce à usiner (23) pour la distance de profondeur (w) au pixel obtenu par lancer de rayon ;
- un point de coordonnées tridimensionnel q (u, v, w) obtenu en combinant les coordonnées de pixel (u, v) et la distance de profondeur (w) est converti dans le système de coordonnées des données de forme de pièce à usiner pour trouver le point de coordonnées tridimensionnel P (x, y, z), et
- un segment de ligne (L) perpendiculaire au plan de projection associé à l'image projetée et commençant au point P dans la direction de profondeur est calculé comme données de rayon.

50

55

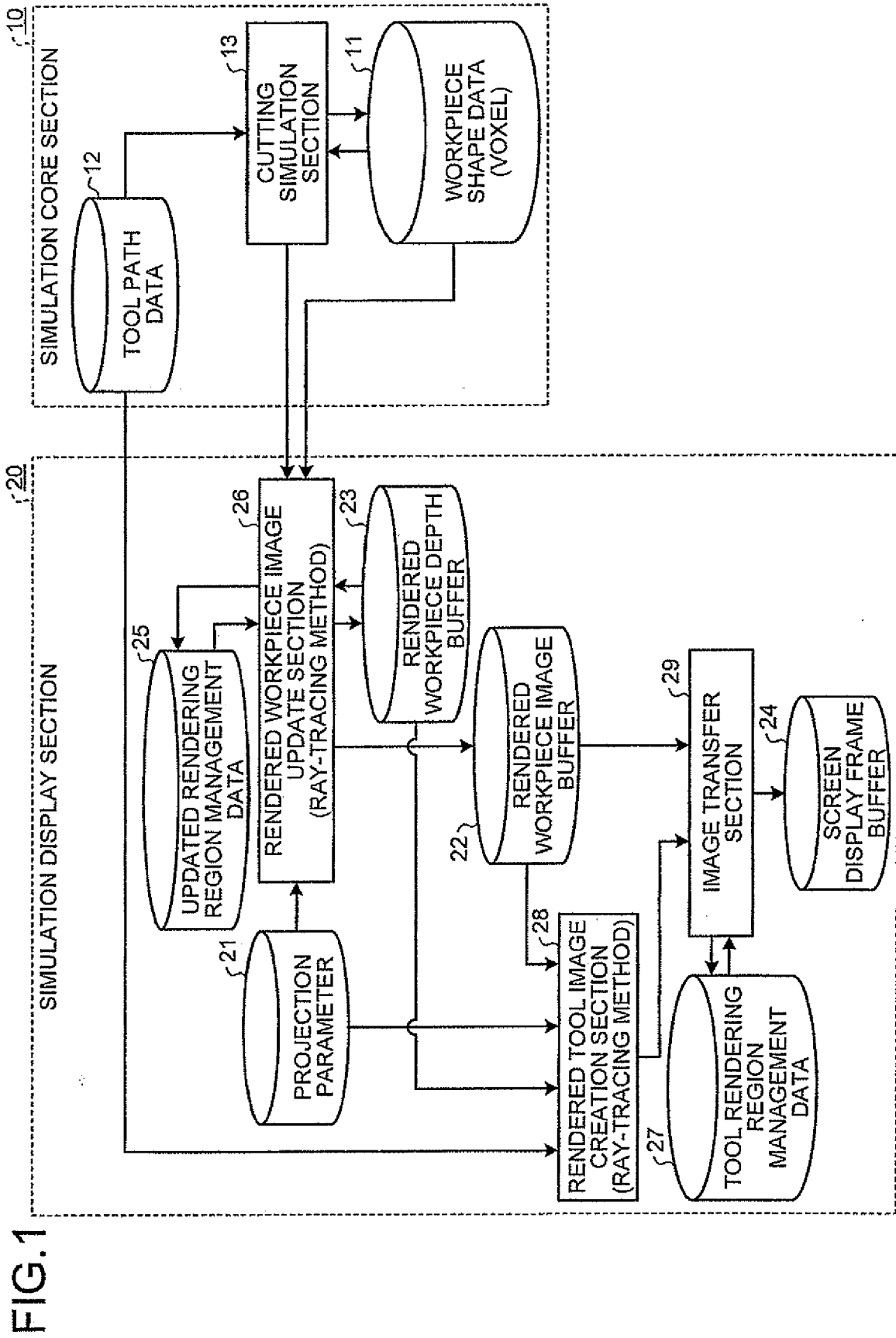


FIG.2-1

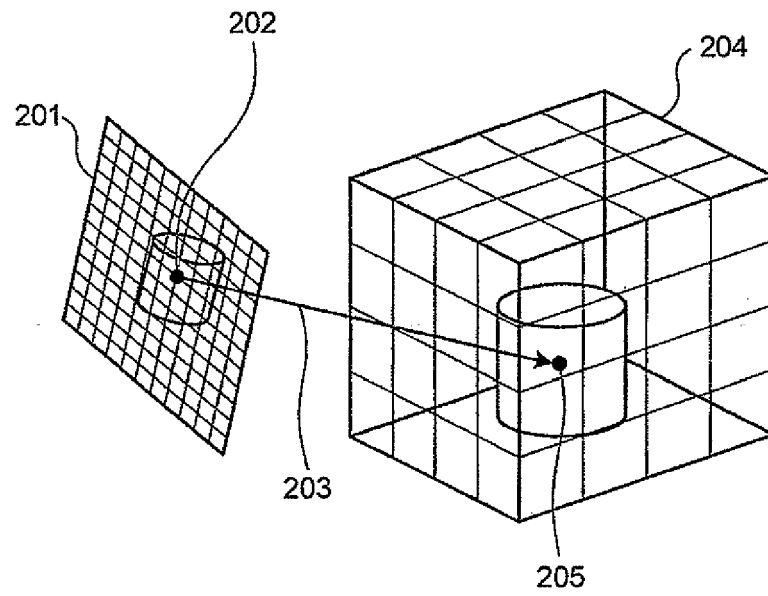


FIG.2-2

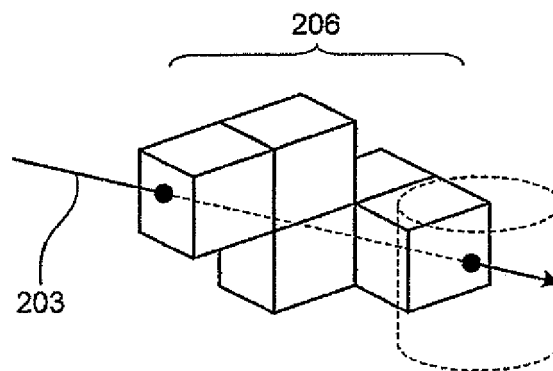


FIG.3

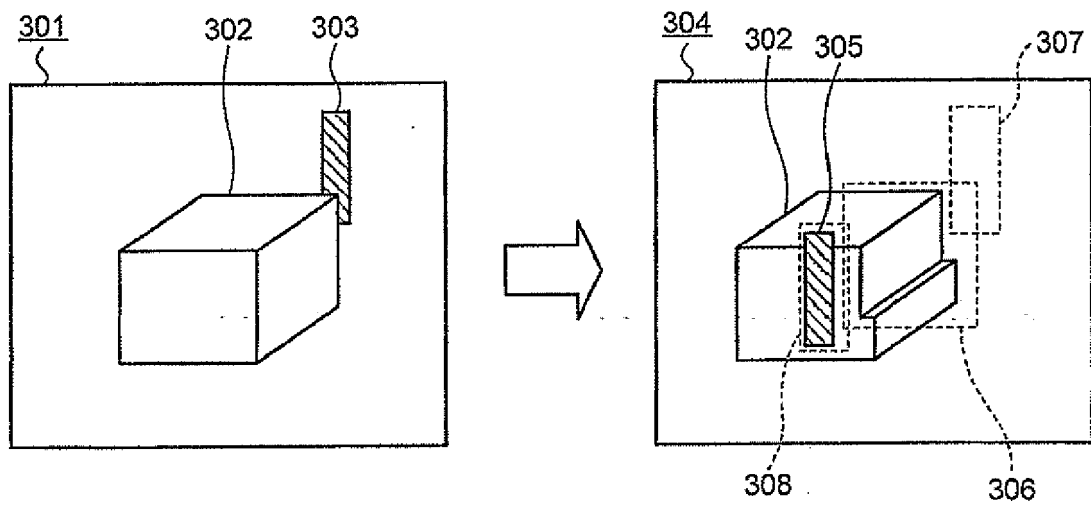


FIG.4-1

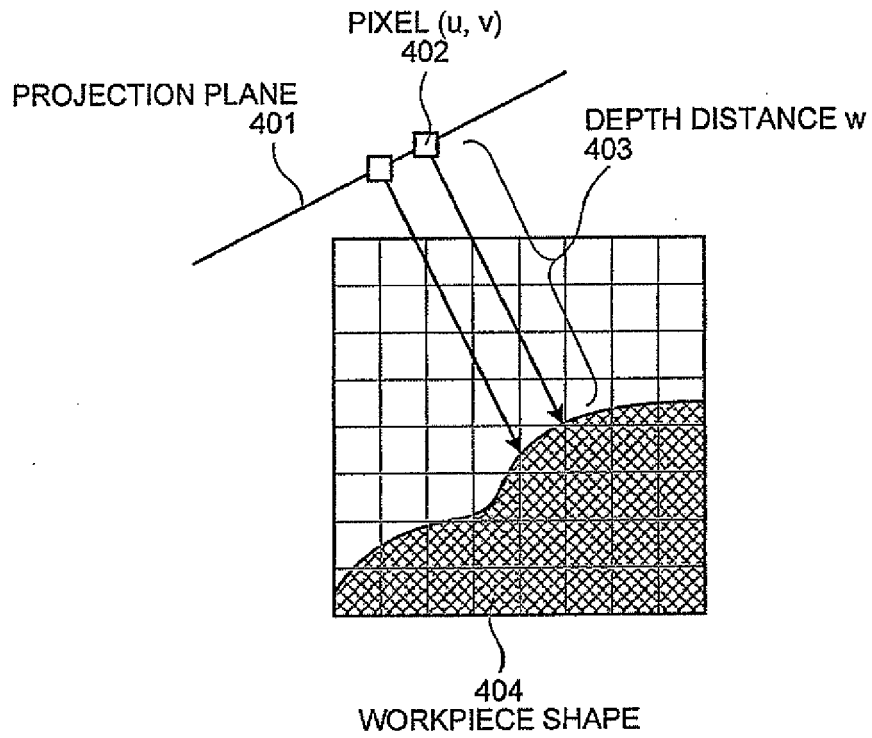


FIG.4-2

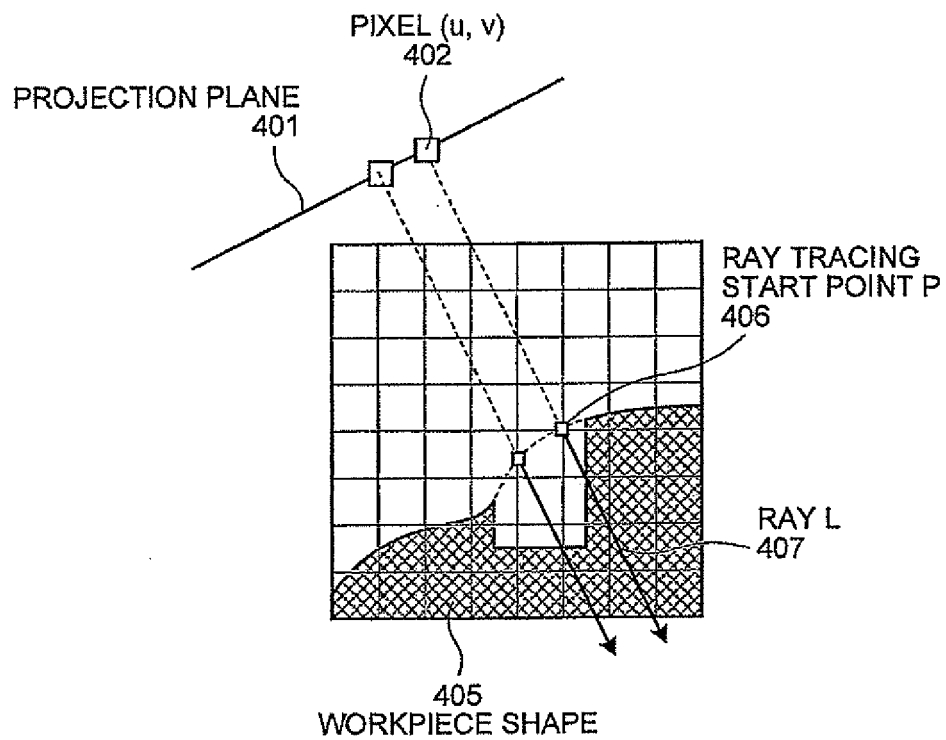


FIG.5

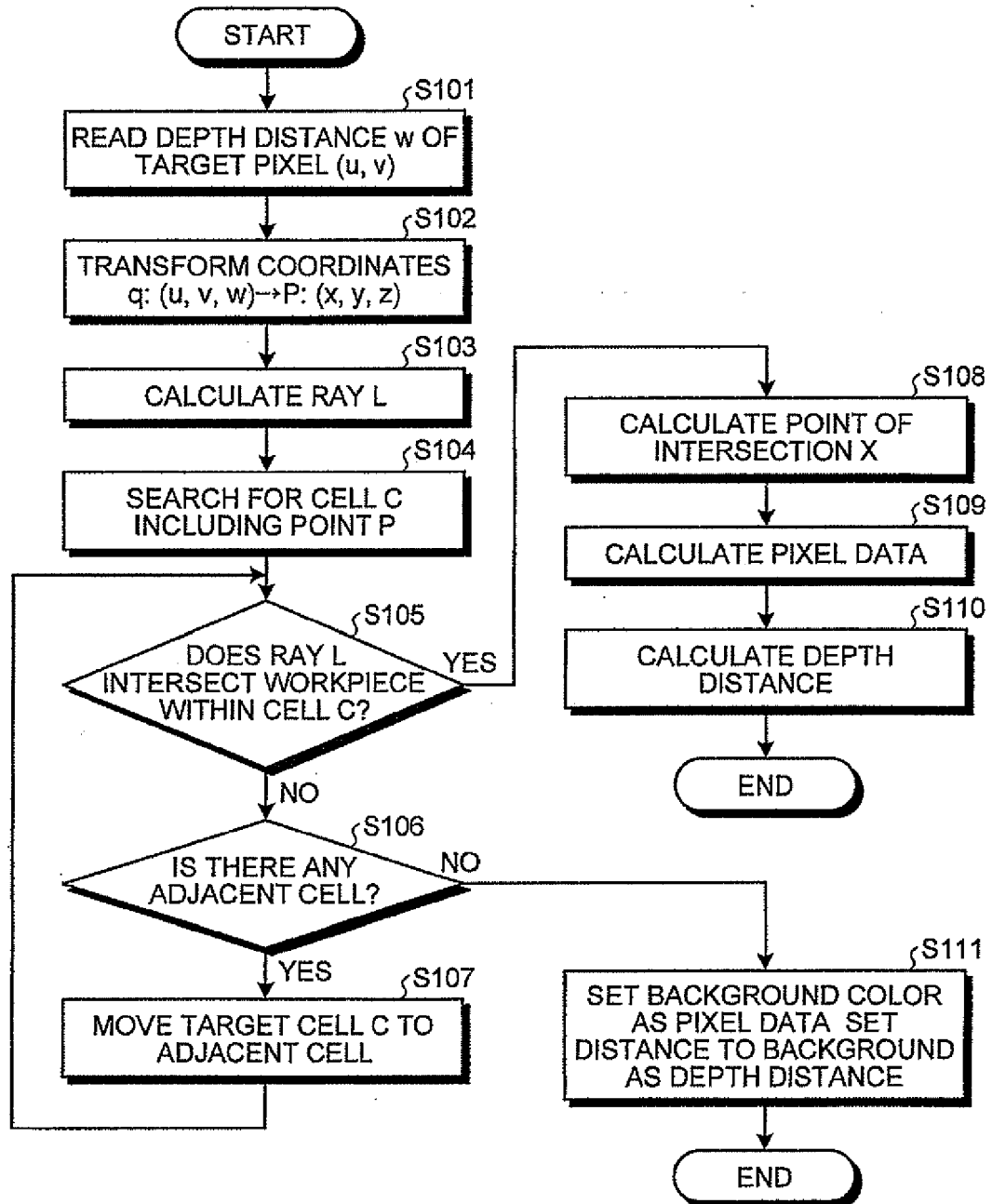
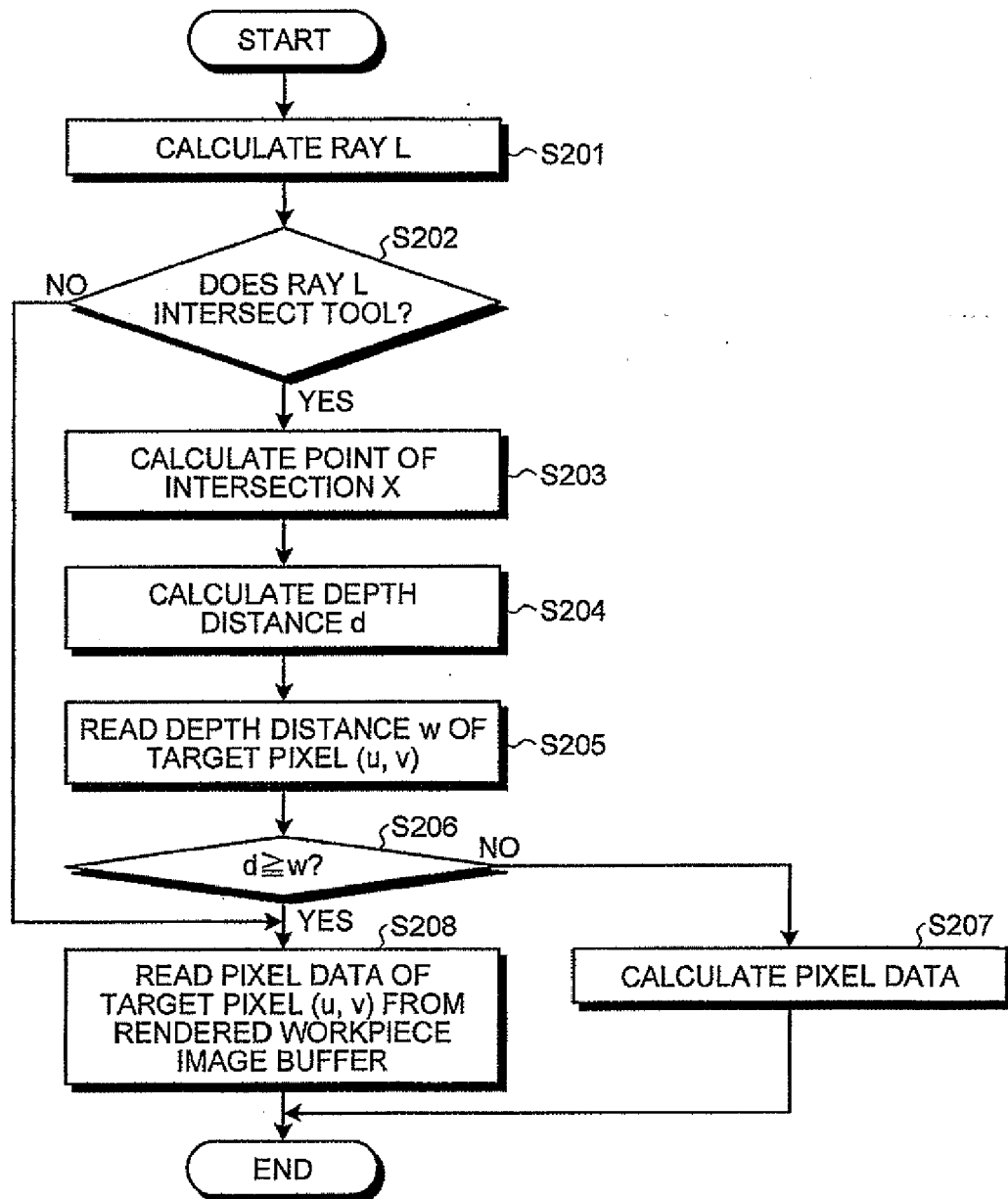


FIG.6



REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 2002133264 A1 [0002]
- JP 3571564 B [0004]

Non-patent literature cited in the description

- **RONI YAGEL et al.** Accelerating volume animation by space-leaping. the Ohio State University Department of Computer and Information Science, 25 October 1993 [0002]