



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 326 755**

51 Int. Cl.:
G06T 15/00 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **06728229 .3**

96 Fecha de presentación : **04.04.2006**

97 Número de publicación de la solicitud: **1866870**

97 Fecha de publicación de la solicitud: **19.12.2007**

54 Título: **Representación de gráficos informáticos 3D utilizando prestaciones 2D de gráficos informáticos.**

30 Prioridad: **04.04.2005 US 667228 P**

45 Fecha de publicación de la mención BOPI:
19.10.2009

45 Fecha de la publicación del folleto de la patente:
19.10.2009

73 Titular/es: **Revolver Ltd.**
22 Maskit Street, P.O. Box 12755
Herzliya 46733, IL

72 Inventor/es: **Fischer, Amir y**
Elber, Gershon

74 Agente: **Carpintero López, Mario**

ES 2 326 755 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Representación de gráficos informáticos 3D utilizando prestaciones 2D de gráficos informáticos.

5 **Campo de la invención**

La presente invención versa acerca del campo de gráficos informáticos tridimensionales (3D). Específicamente, la presente invención versa acerca de la generación y/o la representación visual de gráficos informáticos 3D, en los que cualquier usuario final puede representar visualmente las imágenes 3D en su dispositivo terminal, por ejemplo, un ordenador personal, un teléfono móvil, etc.

Antecedentes de la invención

La infografía 3D es el arte de utilizar ordenadores digitales y *software* 3D especializado para crear una colección de objetos gráficos que pueden ser representados visualmente *en un terminal adecuado*. El documento WO 2004/036504 (Nokia Corp.) desvela un procedimiento para preprocesar gráficos 3D en un primer dispositivo con una mayor potencia de procesamiento y de representar gráficos animados 3D en un segundo dispositivo con un procesamiento de potencia relativamente baja. En la etapa de preprocesamiento se generan tiras triangulares de forma que las superficies ocultas se borran y se utilizan los planos de simetría para borrar puntos correspondientes simétricamente para reducir aún más el número total de puntos, que tienen que ser transferidos al segundo dispositivo. Los gráficos informáticos 3D son distintos a los gráficos informáticos 2D en el sentido de que la representación 3D de los datos geométricos se almacena en un ordenador con el fin de llevar a cabo cálculos y ejecutar instrucciones, requeridos para representar una colección de imágenes 2D que son representadas visualmente y manipuladas, de forma que se obtiene una ilusión 3D.

El procedimiento convencional de crear gráficos informáticos 3D puede dividirse secuencialmente en cinco etapas básicas:

1. modelado
2. configuración de la composición de la escena;
3. transformación;
4. eliminación de superficie oculta; y
5. sombreado y relleno de polígono(s), por ejemplo, mediante un mapeo de superficie.

Se puede describir la etapa de modelado como el moldeo de objetos individuales que son utilizados más adelante en la escena. Se conocen varias técnicas de modelado en la técnica, incluyendo modelado poligonal (se puede encontrar información acerca de modelado poligonal, por ejemplo, en http://en.wikipedia.org/wiki/Polygonal_modeling). Se pueden crear modelos con una amplia variedad de herramientas comerciales de modelado, como AutoCAD[®] y Solid Works[®].

Los polígonos son figuras simples utilizadas por virtualmente toda herramienta de representación como sus figuras básicas. Normalmente, los polígonos son aproximados desde otras representaciones geométricas como superficies para animación (curvadas). Entonces, normalmente se convierten las escenas en polígonos, normalmente triángulos. Como comprenderán quienes estén familiarizados con la técnica de los gráficos 3D, a un grupo de polígonos que están conectados entre sí por medio de vértices compartidos se lo denomina una malla. La ventaja principal de la representación poligonal es que es más eficaz que otros tipos de representaciones gráficas para las necesidades de representación y de creación de imágenes.

Una vez se han representado todos los objetos mediante polígonos en la orientación correcta (es decir, la orientación, en la que el usuario verá la escena deseada) deben ser representados visualmente en la pantalla del ordenador. El ojo humano ve tres dimensiones, mientras que la pantalla del ordenador únicamente puede mostrar dos dimensiones, por lo tanto, el modelo 3D debe ser convertido a una o más imágenes bidimensionales. Esto se lleva a cabo habitualmente utilizando proyección, preferentemente proyección en perspectiva. La idea básica tras la proyección perspectiva es similar a la forma en la que trabaja el ojo humano. Una vez se ha proyectado la imagen sobre la pantalla, los objetos más lejanos son más pequeños con respecto a los que están más cercanos al ojo, de esta manera se crea una ilusión de 3D.

La etapa de configuración de la composición de la escena implica disponer objetos virtuales, luz, sombreado y otras entidades sobre una escena que será utilizado más adelante para producir una imagen fija o una animación. La iluminación es un factor importante en una configuración de escena. Como en el caso de la disposición de la escena en el mundo real, la iluminación es un factor contribuyente significativo a la calidad estética y visual resultante del efecto visual final.

Durante la etapa de eliminación de la superficie oculta, se aplica un análisis de visibilidad para determinar y mostrar únicamente las porciones visibles de la imagen 3D creada. Normalmente, el análisis de visibilidad se lleva a

cabo utilizando un búfer Z, que es una solución al problema (problema de la visibilidad) de decidir qué elementos de una escena son visibles, y cuáles están ocultos (se puede encontrar información acerca de búferes z, por ejemplo, en <http://en.wikipedia.org/wiki/Z-buffering>). Prácticamente, el almacenamiento temporal Z es la gestión de las coordenadas de profundidad de las imágenes en gráficos 3D.

Cuando se representa un objeto mediante una tarjeta gráfica 3D, se almacena la profundidad de un píxel generado (coordenada z) en un búfer (el búfer Z). Normalmente, este búfer está dispuesto como un conjunto bidimensional (x-y) con un elemento por cada píxel de la pantalla. Si se tiene que representar otro objeto de la escena en el mismo píxel, la tarjeta gráfica compara los dos valores de profundidad y selecciona el que esté más cercano al observador. Entonces, se guarda el valor escogido de profundidad en el búfer Z, sustituyendo el anterior. Al final, el búfer Z permitirá que la tarjeta gráfica reproduzca correctamente la percepción normal de profundidad: un objeto más cercano oculta uno más lejano.

Sin embargo, la implementación de la técnica del búfer Z en *software* puede deteriorar las prestaciones inherentes de las herramientas gráficas instaladas en el ordenador. Por ejemplo, un paquete gráfico 2D puede soportar un dibujo muy eficaz de polígonos 2D, incluyendo un relleno rápido y un mapeo de superficie eficaz. Al utilizar una emulación de software del búfer Z, se pueden deteriorar, o incluso perder, esas prestaciones. También se puede utilizar el bien conocido algoritmo del pintor, que es una de las soluciones más sencillas al problema de la visibilidad en gráficos informáticos 3D, para el análisis de visibilidad (se puede encontrar información acerca del algoritmo del pintor, por ejemplo, en http://en.wikipedia.org/wiki/Painter's_algorithm). El algoritmo del pintor clasifica todos los polígonos en una escena mediante sus profundidades y luego los pinta en este orden, comenzando con el polígono más lejano hasta el más cercano. Sobrepintará las partes que no son visibles normalmente y de esta manera soluciona el problema de la visibilidad.

Haciendo referencia a dos polígonos arbitrarios P y Q, el algoritmo clásico del pintor (descrito, por ejemplo, en "Computer Graphics: Principles and Practice", por Foley *et al*, Addison-Wesley, 1996), lleva a cabo el análisis de visibilidad conforme a las siguientes pruebas:

- a. Probar si las extensiones-x (los valores x que se extienden por los polígonos) de los dos polígonos no se solapan. Si es así, el orden es irrelevante.
- b. Probar si las extensiones-y de los dos polígonos no se solapan. Si es así, el orden es irrelevante.
- c. Probar si P está completamente a un lado del plano de Q (es decir, P no intersecta a Q). Si es así, se puede determinar el orden.
- d. Probar si Q está completamente a un lado del plano de P (es decir, Q no intersecta a P). Si es así, se puede determinar el orden.
- e. Probar si las proyecciones de P y Q sobre el plano XY no se solapan. Si es así, el orden es irrelevante.

Una vez se ha definido un orden parcial para cada par de polígonos en la escena, se debe determinar un orden del conjunto completo, de forma que el pintado del conjunto de polígonos en ese orden dejará una imagen 2D adecuada de la escena con las porciones ocultas borradas. Dicho orden podría ser extraído de la información en todos los pares de polígonos utilizando, por ejemplo, un procedimiento de búsqueda de gráficos.

Sin embargo, el algoritmo del pintor puede fallar en ciertos casos. La Fig. 1 ilustra una situación en la que los polígonos 1, 2 y 3 se solapan entre sí. La Fig. 4 ilustra otra situación similar, relacionada con los polígonos P y Q. En ambos casos, no es posible decidir qué polígono está más cerca que los otros. Para implementar el algoritmo del pintor en dichos casos, los polígonos que se solapan deben estar divididos de alguna forma en subpolígonos, de forma que se pueda determinar el orden de los subpolígonos.

Hasta ahora hemos tratado el algoritmo del pintor que soluciona el problema de la visibilidad para un conjunto de polígonos para una única dirección de visión. Una extensión del algoritmo del pintor que intenta solucionar los problemas de visibilidad desde todas las vistas posibles es conocida en la técnica como Partición binaria del espacio (BSP) (se puede encontrar información acerca de BSP, por ejemplo, en http://en.wikipedia.org/wiki/Binary_space_partitioning). Una de las desventajas del procedimiento BSP es que requiere la subdivisión de muchos polígonos, y, por lo tanto, aumenta enormemente el número total de polígonos que deben ser procesados.

Los gráficos informáticos 3D se han vuelto más y más dominantes en muchos campos relacionados con los ordenadores, incluyendo simulaciones científicas, películas de animación, y juegos informáticos. Por lo tanto, ha estado creciendo continuamente la necesidad de un *hardware* y un *software* gráficos 3D.

Algunos ordenadores modernos comprenden *hardware* y *software* capaz de presentar y/o generar gráficos 3D, sin embargo, esto no es cierto para todos los ordenadores que están generalizados actualmente en el mercado. Además muchas aplicaciones son inherentemente bidimensionales, por ejemplo, la tecnología utilizada habitualmente Macromedia® Flash. Dichas tecnologías 2D no soportan la transformación 3D ni la proyección 3D a 2D. Por ejemplo, la rotación de un objeto, por ejemplo, un cubo, mientras que se lo observa desde todas las direcciones, puede requerir

la conversión de geometría 3D a geometría 2D, una característica que no está disponible en paquetes informáticos gráficos 2D. Además, los paquetes informáticos gráficos 2D son incapaces de resolver el problema de la eliminación de superficies ocultas.

5 Los juegos informáticos tienden a requerir gráficos informáticos 3D, al igual que muchos recursos del ordenador. Esto también es cierto para distintas aplicaciones 3D que son ejecutadas en otros dispositivos terminales, por ejemplo, teléfonos móviles.

10 Una forma de solucionar el problema presentado anteriormente en el presente documento es implementar lo que se conoce en la técnica como “módulos de ampliación”. Los módulos de ampliación son paquetes informáticos que son descargados normalmente de internet para ampliar las prestaciones del ordenador. Una vez descargados, los módulos de ampliación se instalan ellos mismos automáticamente, de forma que los usuarios no necesitan tener conocimiento técnico para utilizarlos. Sin embargo, la descarga de paquetes informáticos de internet puede poner en riesgo la seguridad del ordenador. No todas las páginas *web* son seguras, y dichas descargas pueden infectar al ordenador con diversos tipos de “virus” y “gusanos” informáticos.

20 Por lo tanto, sería altamente deseable permitir a cualquier usuario final utilizar y disfrutar de gráficos 3D, por ejemplo en juegos informáticos, sin la necesidad de descargar módulos de ampliación y sin poner en peligro el dispositivo terminal de cualquier otra forma.

Es un objeto de la presente invención proporcionar un procedimiento mediante el cual cualquier usuario final puede gestionar gráficos informáticos 3D en cualquier dispositivo terminal que comprende aplicaciones gráficas 2D convencionales.

25 Es otro objeto de la presente invención proporcionar un procedimiento que proporciona de manera remota a cualquier usuario final la capacidad de gestionar gráficos informáticos y animaciones 3D, en internet.

Es aún otro objeto de la presente invención proporcionar un procedimiento que permite a cualquier usuario final gestionar gráficos informáticos 3D sin deteriorar la seguridad de su dispositivo terminal.

30 Es aún otro objeto de la presente invención proporcionar un procedimiento que permite a cualquier usuario final gestionar gráficos informáticos 3D sin sobrecargar el canal de datos mediante el que el usuario está conectado a internet.

35 Es un objeto adicional de la presente invención proporcionar un procedimiento que permite a cualquier usuario final gestionar gráficos informáticos 3D utilizando el intercambio existente de tasa de datos de su dispositivo terminal.

Es aún otro objeto adicional de la presente invención proporcionar un procedimiento eficaz para llevar a cabo un análisis de visibilidad.

40 Serán evidentes fines y ventajas adicionales de la presente invención según avance en la descripción.

Resumen de la invención

45 La presente invención está dirigida a un procedimiento para generar y mostrar imágenes gráficas 3D en la pantalla de un dispositivo terminal utilizando entornos gráficos 2D, en el que cada escena posible está modelada mediante uno o más objetos que pueden ser representados mediante polígonos utilizando un procedimiento adecuado de modelado. Se proyecta la geometría de los polígonos que se corresponden a cada objeto para cualquier orientación deseada de los objetos, sobre el plano de la pantalla. Para cada objeto, se crean curvas que conectan las proyecciones de los vértices de su malla correspondiente de polígonos en todas las distintas orientaciones de ese objeto y que consisten en una pluralidad de puntos discretos, de forma que cada punto en cada curva está almacenado conforme a su posición (x , y) en el plano, y de forma que la resolución de cada curva está determinada conforme al número de puntos. Se lleva a cabo un análisis de la visibilidad para cada malla que se corresponde con una orientación específica, determinando de ese modo la distancia de los puntos desde el observador. Se borran los polígonos y/o las porciones o los bordes ocultos y se codifica de manera óptima la geometría para todas las orientaciones, por ejemplo, al utilizar un Árbol de extensión mínima. Entonces, se muestran imágenes gráficas 3D en los entornos 2D al utilizar la geometría codificada para reconstruir una porción de, o todos, los polígonos restantes y rellenar los polígonos restantes conforme a las reglas predefinidas. El relleno puede incluir un mapeo de superficie y/o pintura.

60 La etapa de codificación puede incluir además datos relacionados con el tiempo (t) en el que cada punto alcanza una posición deseada.

Preferentemente, el análisis de visibilidad se lleva a cabo para los polígonos al probar si las extensiones- x de los dos polígonos no se solapan, y si lo hacen, marcar los dos polígonos como polígonos de los que el orden de representación visual es irrelevante; en caso contrario, probar si las extensiones- y de los dos polígonos no se solapan, y si lo hacen, marcar los dos polígonos de los que el orden de representación visual es irrelevante; en caso contrario, probar si las proyecciones de ambos polígonos sobre el plano de la vista no se solapan, y si lo hacen, marcar los dos polígonos como polígonos de los que el orden de representación visual es irrelevante; en caso contrario, probar si el

primer polígono está completamente a un lado del plano del segundo polígono, y si lo hacen, determinar el orden de representación visual de los dos polígonos; en caso contrario, probar si el segundo polígono está completamente a un lado del plano del primer polígono, y si lo hacen, determinar el orden de representación visual de los dos polígonos; en caso contrario, probar si existe un plano de separación entre los dos polígonos, y si lo hacen, determinar el orden de representación visual de los dos polígonos; en caso contrario, particionar polígonos según sea necesario. Estos pasos se repiten con los polígonos particionados. Se encuentra el orden apropiado entre todos los polígonos en la escena al realizar una búsqueda de gráficos por todos los resultados de los pasos precedentes de todos los pares posibles de polígonos, mientras que se particionan polígonos según sea necesario. Entonces, se repiten los mismos pasos para cada par de polígonos en cada orientación.

El análisis de visibilidad puede llevarse a cabo para al menos dos polígonos convexos al: probar si las extensiones- x de los dos polígonos convexos no se solapan, y si lo hacen, marcar los dos polígonos como polígonos de los que el orden de representación visual es irrelevante; en caso contrario,

probar si las extensiones- y de los dos polígonos no se solapan, y si lo hacen, marcar los dos polígonos convexos como polígonos de los que el orden de representación visual es irrelevante; en caso contrario,

probar si las proyecciones de ambos polígonos convexos sobre el plano de la vista no se solapan, y si lo hacen, marcar los dos polígonos convexos como polígonos de los que el orden de representación visual es irrelevante; en caso contrario

probar si el primer polígono convexo está completamente a un lado del plano del segundo polígono convexo, y si lo hacen, determinar el orden de representación visual de los dos polígonos convexos; en caso contrario, probar si el segundo polígono convexo está a un lado del plano del primer polígono convexo, y si lo hacen, determinar el orden de representación visual de los dos polígonos convexos; en caso contrario, encontrar un plano de separación entre los dos polígonos convexos y determinar el orden de representación visual de los dos polígonos convexos;

encontrar el orden apropiado entre todos los polígonos convexos en la escena al llevar a cabo una búsqueda de gráficos por todos los resultados de los pasos (a) a (f) de todos los pares posibles de polígonos convexos, mientras es necesario particionar polígonos convexos,

en el que se repiten los pasos (a) a (f) para cada par de polígonos convexos en todas las orientaciones.

Preferentemente, se prueba la existencia de un plano de separación conforme a los siguientes pasos:

- a. definir un plano L como un plano a través del borde e_i y (vértice) $_j$ en el que los bordes e_i son los bordes que conectan el (vértice) $_j$ y el (vértice) $_{i+1}$ del primer polígono y el (vértice) $_j$ es un vértice del segundo polígono;
- b. comprobar si el primer polígono se encuentra a un lado de L y el segundo polígono se encuentra al otro lado de L y siempre que el resultado sea negativo;
- c. repetir el paso (b) para todas las combinaciones entre los bordes e_i y los vértices (vértice) $_j$ hasta que se encuentre un plano de separación; en caso contrario,
- d. definir el plano L como un plano a través del borde e_j y el (vértice) $_i$ en el que los bordes e_j son los bordes que conectan el (vértice) $_i$ y el (vértice) $_{j+1}$ del segundo polígono y el (vértice) $_i$ es un vértice del primer polígono;
- e. comprobar si el primer polígono se encuentra a un lado de L y el segundo polígono se encuentra al otro lado de L , y, siempre que el resultado sea negativo;
- f. repetir el paso (e) para todas las combinaciones entre los bordes e_j y los vértices (vértice) $_i$ hasta que se encuentre un plano de separación o hasta que se hayan comprobado todas las combinaciones.

La geometría puede estar codificada utilizando un árbol de extensión mínima, de forma que se generan n listas ordenadas $\{Ti_{i=1,\dots,n}\}$ de subconjuntos de la malla poligonal que representa cada vista para cualesquiera n vistas distintas deseadas. También se genera un gráfico clicable que consiste en nodos n (N_i) siendo las n listas ordenadas Ti y en bordes Eij que conectan el nodo i con el nodo j $_{i=1,\dots,n}$. Se asigna un peso que se corresponde con el coste de la diferencia entre cualesquiera dos nodos N_i y N_j para cada borde Eij en el gráfico clicable y entonces se genera el Árbol de extensión mínima (MST) del gráfico clicable. Todas las listas ordenadas $\{Ti\}$ se representan mediante una lista de orden seleccionado Tk , es decir, la raíz del MST, y mediante todos los bordes seleccionados con el peso mínimo que son parte del MST, y yendo desde Tk hasta todas las listas ordenadas restantes. Tk y todas las transiciones representadas por los bordes seleccionados están codificadas en un fichero de datos, a partir del que se pueden reconstruir todas las n listas Ti ordenadas.

Se comprenderán mejor todas las anteriores y otras características y ventajas de la invención mediante la siguiente descripción ilustrativa y no limitante de las realizaciones preferentes de la misma.

Breve descripción de los dibujos

■ La Fig. 1 muestra esquemáticamente tres polígonos que se solapan;

5 ■ Las Figuras 2a, 2b, 3a y 3b describen esquemáticamente un cubo giratorio y las curvas creadas a partir del movimiento de sus vértices; y

■ La Figura 4 describe esquemáticamente dos polígonos que se solapan en el espacio 3D.

10 **Descripción detallada de las realizaciones preferentes**

En vista de los inconvenientes de la técnica anterior, como se han descrito anteriormente en el presente documento, existe la necesidad de un procedimiento mejorado mediante el que los usuarios puedan ser provistos de gráficos 3D, o con la capacidad para generar gráficos 3D de una forma eficaz en entornos que solo soportan gráficos 2D.

15 En una realización de la presente invención, se proporciona una página *web* segura que ofrece contenido 3D, como juegos informáticos y animaciones. Cualquier usuario final, incluso uno que no posee paquetes gráficos 3D, suscrito a la página *web* puede jugar a los juegos informáticos ofrecidos mientras que está conectado a la página *web* durante el tiempo de juego. A diferencia de los riesgos causados al descargar módulos de ampliaciones, la página *web* implementada en esta realización de la presente invención es segura, por lo tanto no hay ningún riesgo de seguridad
20 al dispositivo terminal, por ejemplo, ordenador o teléfono móvil, mientras que el usuario se encuentra conectado a la página *web*.

En otras realizaciones de la presente invención, se le puede facilitar a cualquier usuario la capacidad de producir
25 gráficos informáticos 3D utilizando las aplicaciones gráficas 2D de su dispositivo terminal, incluso cuando no está conectado a la página *web* mencionada anteriormente.

Preferentemente, los gráficos 3D utilizados en la presente invención están generados en una etapa de preprocesamiento sin conexión. Conforme a la presente invención, las imágenes que van a ser utilizadas en cualquier escena
30 3D son normalmente modelados utilizando mallas poligonales, como se ha descrito anteriormente en el presente documento. Una vez se han generado los polígonos, se proyectan sobre el campo de visión, (es decir, el plano de la pantalla del dispositivo terminal). Esta proyección se lleva a cabo para un conjunto discreto de vistas de los objetos de cualquier escena durante toda la presentación. Al resolver la cuestión de la visibilidad para únicamente un conjunto discreto de vistas, se ideó un plan más eficaz, con respecto al número de particiones de polígonos que son necesarios
35 en comparación con otros procedimientos, como la BSP.

Para paliar las dificultades que provienen del hecho de que solo se emplea un conjunto discreto de vistas, se representaron gráficamente en el plano de representación visual cambios en la posición de los vértices de cada polígono, resultantes de todas las orientaciones muestreadas, y se trazan las curvas que conectan las distintas posiciones del
40 mismo vértice. La resolución de los cambios, representados gráficamente en aquellas curvas, depende del número de orientaciones muestreadas cuando se producen. Dado que las curvas se generan mediante el movimiento de los vértices con el tiempo, en una realización de la presente invención cada punto de cada curva puede ser almacenado en una matriz conforme a su posición (x, y) y opcionalmente conforme al tiempo (t). Es decir, la memoria del dispositivo terminal contiene información relacionada con las posiciones de todos los objetos en cualquier escena, además de su
45 velocidad de movimiento. En otras realizaciones, solo se almacenan las posiciones (x, y) en la memoria del dispositivo terminal. En aquellas realizaciones, el valor de tiempo está definido mediante otras reglas predefinidas del juego, y/o por las acciones del usuario final.

Se describe un ejemplo de dichas curvas en las Figuras 2a, 2b, 3a y 3b. Todas esas figuras describen un cubo (20)
50 que gira. En la Fig. 2a se describe el cubo (20) como que gira 45 grados en contra de las agujas del reloj en torno al eje vertical (22). Observando, por ejemplo, el vértice en la posición (30) en la Fig. 2a, se puede ver que cuando el cubo (20) gira 45 grados en torno al eje (22), el vértice en la posición (30) se mueve por la curva (31), y alcanza la posición (32). De la misma forma, los vértices en las posiciones (35), (40), (45), (50), (55) y (60) se mueven por las curvas (36), (41), (46), (51), (56) y (61) hasta sus nuevas posiciones (37), (42), (47), (52), (57) y (62), respectivamente.

55 Cuando el cubo (20) gira 45 grados en el sentido de las agujas del reloj en torno al eje (22), como se puede ver en la Fig. 2b, el vértice en la posición (30) se mueve por la curva (33) y alcanza la posición (34). De la misma forma, los vértices en las posiciones (35), (40), (45), (50), (55) y (60) se mueven por las curvas (38), (43), (48), (53), (58) y (63) hasta sus nuevas posiciones (39), (44), (49), (54), (59) y (64), respectivamente.

60 Las Figuras 3a y 3b describen el mismo tipo de movimiento rotatorio, aunque en estas figuras el cubo (20) gira en torno a un eje horizontal (24). Cuando gira 135 grados apartándose del observador, como se describe en la Fig. 3a, el vértice en la posición (30), por ejemplo, se mueve por la curva (33') hasta que alcanza la posición (34'). De la misma forma, los vértices (50), (55) y (60) se mueven por las curvas (53'), (58') y (63') hasta sus nuevas
65 posiciones (54'), (59') y (64'), respectivamente. Los vértices en las posiciones (35), (40) y (45) se mueven hasta las posiciones directamente bajo las posiciones (54'), (59') y (64'), por lo tanto, su movimiento no se muestra en el dibujo.

En la Fig. 3b el cubo (20) se describe como que gira 45 grados hacia el observador en torno a un eje horizontal (24). Como se puede ver en la Fig. 3b, el vértice en la posición (30), por ejemplo, se mueve por la curva (31') hasta que alcanza la posición (32'). De la misma forma, los vértices (35), (40) y (45) se mueven por las curvas (36'), (41') y (46') hasta sus nuevas posiciones (37'), (42') y (47'), respectivamente. Los vértices en las posiciones (50), (55) y (60) se mueven hasta las posiciones directamente bajo las posiciones (37'), (42') y (47'), por lo tanto, su movimiento no se muestra en el dibujo.

Se debería comprender que el número de estados, es decir, las posiciones generadas y almacenadas en memoria durante el preprocesamiento, es finito. Sin embargo, una vez se le suministran al usuario los valores discretos (x , y) (y posiblemente (t)), su dispositivo terminal puede calcular simplemente cualquier estado intermedio deseado, por ejemplo mediante interpolación o extrapolación.

Una vez se han generado las curvas mencionadas anteriormente, se lleva a cabo un análisis de visibilidad para cada punto (x , y), por ejemplo, las posiciones (30) y (32) como se muestran en la Fig. 2a. Si se almacenó el tiempo (t) en memoria al igual que la posición (x , y), el análisis visual será llevado a cabo para cada punto (x , y) en cada tiempo (t). En una realización preferida de la presente invención, el análisis de profundidad se lleva a cabo utilizando una modificación única del algoritmo del pintor que casi elimina la necesidad de divisiones poligonales.

La Fig. 4 muestra polígonos P y Q de la dirección +Z en la que ambos polígonos son casi verticales. Cuando se examinan los polígonos P y Q conforme a las cinco pruebas del algoritmo clásico del pintor descritas anteriormente en el presente documento, es evidente que fallarán las cinco pruebas. Los polígonos P y Q se solapan tanto en su extensión-x (es decir, el dominio-x que abarca el polígono P se intersecta con el dominio-x que abarca el polígono Q) y en su extensión-y, por lo tanto, las pruebas 1 y 2 fallarán. Además, dado que ambos polígonos son casi verticales, el plano de P intersecta Q, y viceversa, así, las pruebas 3 y 4 también fallarán. Finalmente, la prueba 5 fallará dado que, como se ve claramente en la Fig. 4, las proyecciones de los polígonos P y Q sobre el plano XY se solapan. Dado que han fallado las cinco pruebas, es necesario, cuando se utiliza el algoritmo clásico del pintor, dividir uno de los polígonos en subpolígonos de forma que se pueda determinar un orden parcial. Solo entonces será posible disponer los polígonos conforme a sus distancias desde el observador (valor de profundidad). Sin embargo, sería altamente deseable minimizar el número de divisiones de polígonos, minimizando de ese modo los recursos informáticos necesarios para el procedimiento del análisis de visibilidad.

En una realización preferida de la presente invención, el número de particiones de polígonos se minimiza al añadir una sexta prueba a las cinco pruebas existentes del algoritmo clásico del pintor. La sexta prueba de la presente invención comprueba si existe un plano L de separación, de forma que P esté ubicado a un lado de L, y Q esté ubicado en el otro. Como comprenderán quienes estén familiarizados con la técnica, si existe dicho plano, la determinación del orden de los polígonos a lo largo de eje Z se vuelve insignificante. El procedimiento de búsqueda del plano L se lleva a cabo conforme a los siguientes pasos:

1. Definir los bordes e_i del polígono P como las líneas que conectan los vértices p_i y p_{i+1} del mismo polígono. Por ejemplo, como se muestra en la Fig. 4, e_1 es el borde que conecta p_1 y p_2 .
2. Definir L como un plano que pasa a través del borde e_i de P y el vértice q_j , que es uno de los vértices del polígono Q.
3. Comprobar si P se encuentra a un lado de L, y Q se encuentra al otro lado. Si es así, se determina el orden de los polígonos, si no, se debería repetir el paso 2 para todas las combinaciones entre los bordes e_i y los vértices q_j .
4. Si ninguno de los planos L generados separa P y Q completamente, definir e_j como los bordes del polígono Q, y repetir los pasos 2-3 para los bordes e_j y los vértices p_i de P hasta que bien se encuentre un plano de separación, o bien se hayan comprobado todas las combinaciones de bordes y vértices.

Como comprenderán quienes estén familiarizados con la técnica, no todo par de polígonos tiene un plano de separación. Sin embargo, es posible demostrar que cada par de polígonos convexos que no se intersectan sí tiene un plano de separación. Sea M dicho plano de separación que en cualquier caso no intersecta los polígonos P ni Q. Mover M por su línea normal hasta que toca uno de los vértices de cualquier polígono, es decir, p_i o q_j . Se alude a ese vértice en el presente documento como v_1 . Fijar M tocando el vértice v_1 y girarlo en torno a ese punto utilizando los dos grados restantes de libertad. Continuar girando M hasta que toque otro vértice, digamos v_2 , en cualquier polígono. Continuar girando M hasta que toque un tercer vértice en cualquier polígono, al que llamamos v_3 . Dos de los tres vértices que toca ahora M, es decir, v_1 , v_2 , v_3 , deben ser adyacentes debido a la convexidad de los dos polígonos. Por lo tanto, dado que M pasa a lo largo del borde de un polígono, y toca un vértice del otro, se garantiza que el anterior procedimiento que prueba todos los bordes de un polígono con un vértice del otro (y viceversa) encuentra este M (transformado), como un plano de separación.

En la realización preferida de la presente invención, se cambia el orden de las pruebas del algoritmo clásico del pintor. Dado que si cualquiera de las pruebas 1, 2 o 5, como se han descrito anteriormente en el presente documento, tiene éxito, el orden de los polígonos es irrelevante, en la realización preferida de la presente invención, las pruebas 1, 2 y 5 se llevan a cabo antes. Solo si fallan, se llevan a cabo las pruebas 3 y 4, como se han descrito anteriormente

en el presente documento. Cuando las 5 pruebas fallan, se lleva a cabo la sexta prueba (buscando un plano de separación) descrita anteriormente en el presente documento. Si todos los polígonos en la escena son convexos y libres de intersecciones, se garantiza que tengan éxito las 6 pruebas entre los pares de polígonos. En otros casos (por ejemplo, con polígonos no convexos), en los que fallan las seis pruebas, los polígonos serán particionados como se ha descrito anteriormente en el presente documento. Entonces, las pruebas se llevan a cabo de manera repetida de la misma forma para los polígonos particionados. Como se ha indicado anteriormente, quienes estén familiarizados con la técnica podrían utilizar el que haya un orden determinado entre cada par de polígonos, mediante una búsqueda de gráficos, para encontrar un orden apropiado de todos los polígonos visibles de forma que el algoritmo del pintor produzca la imagen correcta. Se debe hacer notar que incluso para polígonos convexos, dicho orden puede suponer la necesidad de particiones poligonales (véase la Figura 1 para un ejemplo tal).

El orden de los polígonos y sus patrones de ocultación en cada escena son guardados en un fichero relativamente pequeño. Suponiendo que se muestren n vistas distintas (que se corresponden con n orientaciones distintas), cada una marcada mediante V_i , en la que $i = 1, 2, \dots, n$. Cada dicha vista está codificada utilizando un cierto subconjunto ordenado de los polígonos en la malla. Se elimina cualquier polígono que se halla completamente oculto en la vista V_i de la lista de polígonos en esa vista. Cuando se consideran dos vistas colindantes, se espera que las listas ordenadas de polígonos en esas dos vistas sean casi similares. Cuando se cambia de una vista a la otra, se borrarán algunos polígonos, otros serán insertados y algunos cambiarán su orden con respecto a su distancia desde el observador. Por lo tanto, se codifican las n vistas distintas al aprovechar completamente esta coherencia espacial.

Dadas n vistas con n listas ordenadas marcadas $\{T_{ij=1,\dots,n}\}$ de subconjuntos de la malla poligonal, computar el coste de la diferencia entre dos listas cualesquiera, T_i y T_j (es decir, la cantidad de espacio requerido para codificar la diferencia). Esta diferencia incluye la eliminación de polígonos (ocultos) que ya no se encuentran en la nueva lista, la inserción de nuevos polígonos y el intercambio de los órdenes de algunos polígonos. Construir ahora un gráfico clicable (un gráfico con bordes desde cada nodo hasta cada otro nodo) cuyos nodos son las listas ordenadas $\{T_i\}$, y cuyos bordes $\{E_{ij}\}$, en las que E_{ij} es el borde desde el nodo i hasta el nodo j . Los bordes $\{E_{ij}\}$ tienen pesos correspondientes según los costes de codificación de eliminar/insertar/intercambiar la transición entre T_i y T_j (el peso de las operaciones de eliminar y de insertar es menor que el peso de intercambiar, dado que el intercambio requiere una codificación de cambios en dos índices, i y j , mientras que las operaciones de eliminar y de insertar requieren una codificación de cambios en un único índice). Entonces, se construye el Árbol de extensión mínima (MST) de este gráfico (se puede encontrar información acerca del Árbol de extensión mínima, por ejemplo, en http://en.wikipedia.org/wiki/Minimum_spanning_tree). La representación MST es la representación óptima de todas las listas $\{T_i\}$ que están codificadas en el fichero.

El decodificador abrirá este árbol MST y abrirá todas las listas $\{T_i\}$, como parte de su etapa de inicialización. Entonces, una vez se selecciona una orientación deseada de vista, V_i , mediante algo de interacción con el observador, se pintan los distintos polígonos, como se listan en T_i , en el orden apropiado mientras que el color y la iluminación se establecen conforme a las reglas predefinidas de sombreado, como sombreado por coseno, utilizando cualquier herramienta gráfica 2D disponible.

Como comprenderán quienes estén familiarizados con la técnica, la descripción de las distintas escenas de cualquier aplicación utilizando las curvas mencionadas anteriormente permite transiciones suaves entre vistas adyacentes utilizando gráficos 2D. Dicha transformación continua también se denomina en la técnica metamorfosis.

La metamorfosis lineal plana es sumamente deseable dado que puede ser acometida fácilmente por la mayoría de dispositivos terminales de gráficos 2D, como Flash[®] de Macromedia. Además, los gráficos 2D requieren, en general, muchos menos recursos informáticos que los gráficos 3D. En esencia, toda la información necesaria se guarda durante el procedimiento de preprocesamiento en ficheros de tamaños pequeños.

Otra ventaja del procedimiento de la presente invención es que todo lo anterior se lleva a cabo durante un procedimiento de preprocesamiento que se realiza sin conexión, de forma que incluso si en algunos casos puede llevar mucho tiempo, solo se lleva a cabo una vez. De esta forma, mientras los usuarios juegan en línea, todos los gráficos se deberían generar a velocidad elevada, utilizando pocos recursos informáticos. En la práctica, los datos generados y almacenados para un número finito de orientaciones discretas permiten al usuario final generar un número infinito de nuevas orientaciones, mientras que se utilizan las herramientas interactivas existentes de la aplicación gráfica 2D convencional ya instalada.

Aunque se han descrito las realizaciones de la presente invención a modo de ilustración, se comprenderá que la invención puede llevarse a cabo con variaciones, modificaciones y adaptaciones, dentro del alcance de las reivindicaciones.

REIVINDICACIONES

1. Un procedimiento para generar y representar visualmente imágenes gráficas 3D en la pantalla de un dispositivo terminal utilizando entornos gráficos 2D, que comprende los pasos de:

- a. modelar toda posible escena mediante uno o más objetos que pueden ser representados mediante polígonos utilizando un procedimiento adecuado de modelado;
- b. proyectar la geometría de los polígonos que se corresponden con cada objeto sobre el plano de dicha pantalla, para cualquier orientación deseada de dichos objetos;
- c. crear curvas para cada objeto que conecten las proyecciones de los vértices de su correspondiente malla de polígonos en todas las distintas orientaciones de ese objeto y que consiste en una pluralidad de puntos discretos, en las que se almacena cada punto en cada curva conforme a su posición (x, y) en dicho plano, de forma que se determina la resolución de cada curva conforme al número de puntos;
- d. llevar a cabo un análisis de visibilidad para cada malla que se corresponde con una orientación específica, para determinar de ese modo la distancia de dichos puntos desde el observador;
- e. borrar polígonos ocultos y/o bordes o porciones de los mismos;
- f. codificar de manera óptima la geometría para todas las orientaciones; y
- g. representar visualmente imágenes gráficas 3D en dichos entornos 2D al utilizar la geometría codificada para reconstruir al menos una porción de los polígonos restantes y rellenar dichos polígonos restantes conforme a las reglas predefinidas.

2. Un procedimiento conforme a la reivindicación 1, en el que el relleno comprende un mapeo de superficie y/o pintura.

3. Un procedimiento conforme a la reivindicación 1, en el que se codifica la geometría utilizando un árbol de extensión mínima.

4. El procedimiento de la reivindicación 1, en el que la etapa de codificación comprende además datos relacionados con el tiempo (t) en el que cada punto alcanza una posición deseada.

5. El procedimiento de la reivindicación 1, en el que se lleva a cabo el análisis de visibilidad para los polígonos, conforme a los siguientes pasos:

- a. probar si las extensiones-x de dichos dos polígonos no se solapan; y si lo hacen, marcar dichos dos polígonos como polígonos de los que el orden de representación visual es irrelevante; en caso contrario,
- b. probar si dichas extensiones-y de los dos polígonos no se solapan; y si lo hacen, marcar dichos dos polígonos como polígonos de los que el orden de representación visual es irrelevante; en caso contrario,
- c. probar si las proyecciones de ambos polígonos sobre el plano de visión no se solapan; y si lo hacen, marcar dichos dos polígonos como polígonos de los que el orden de representación visual es irrelevante; en caso contrario,
- d. probar si el primer polígono se encuentra completamente a un lado del plano del segundo polígono; y si lo hacen, determinar el orden de representación visual de dichos dos polígonos; en caso contrario,
- e. probar si el segundo polígono se encuentra completamente a un lado del plano del primer polígono, y si lo hacen, determinar el orden de representación visual de dichos dos polígonos; en caso contrario,
- f. probar para ver si existe un plano de separación entre dichos dos polígonos; y si lo hacen, determinar el orden de representación visual de dichos dos polígonos; en caso contrario,
- g. particionar los polígonos según sea necesario;
- h. repetir los pasos (a) a (g) con los polígonos particionados; y
- i. encontrar el orden apropiado entre todos los polígonos en la escena al llevar a cabo una búsqueda gráfica en todos los resultados de (a) a (h) de todos los pares posibles de polígonos, mientras que se particionan polígonos según sea necesario,

en el que se repiten los pasos (a) a (h) para cada par de polígonos en cada orientación.

ES 2 326 755 T3

6. El procedimiento de la reivindicación 1, en el que se lleva a cabo el análisis de visibilidad para al menos dos polígonos convexos conforme a los siguientes pasos:

- a. probar si dichas extensiones-x de los dos polígonos convexos no se solapan, y si lo hacen, marcar dichos dos polígonos como polígonos de los que el orden de representación visual es irrelevante; en caso contrario,
- b. probar si dichas extensiones-y de los dos polígonos convexos no se solapan, y si lo hacen, marcar dichos dos polígonos convexos como polígonos de los que el orden de representación visual es irrelevante; en caso contrario,
- c. probar si las proyecciones de ambos polígonos convexos sobre el plano de visión no se solapan, y si lo hacen, marcar dichos dos polígonos convexos como polígonos de los que el orden de representación visual es irrelevante; en caso contrario,
- d. probar si el primer polígono convexo se encuentra completamente a un lado del plano del segundo polígono convexo, y si lo hacen, determinar el orden de representación visual de dichos dos polígonos convexos; en caso contrario,
- e. probar si el segundo polígono convexo se encuentra completamente a un lado del plano del primer polígono convexo, y si lo hacen, determinar el orden de representación visual de dichos dos polígonos convexos; en caso contrario,
- f. encontrar un plano de separación entre dichos dos polígonos convexos y determinar el orden de representación visual de dichos dos polígonos convexos;
- g. encontrar el orden apropiado entre todos los polígonos convexos en la escena al llevar a cabo una búsqueda gráfica en todos los resultados de los pasos (a) a (f) de todos los pares posibles de polígonos convexos, mientras que se particionan polígonos convexos según sea necesario,

en el que se repiten los pasos (a) a (f) para cada par de polígonos convexos en cada orientación.

7. El procedimiento de la reivindicación 5, en el que se prueba la existencia de un plano de separación conforme a los siguientes pasos:

- a. definir un plano L como un plano a través del borde e_i y del (vértice) $_j$ en el que los bordes e_i son los bordes que conectan el (vértice) $_j$ y el (vértice) $_{i+1}$ del primer polígono y el (vértice) $_j$ es un vértice del segundo polígono;
- b. comprobar si dicho primer polígono se encuentra a un lado de L y dicho segundo polígono se encuentra al otro lado de L y siempre que el resultado sea negativo;
- c. repetir el paso (b) para todas las combinaciones entre los bordes e_i y los vértices (vértice) $_j$ hasta que se encuentre un plano de separación; en caso contrario,
- d. definir un plano L como un plano a través del borde e_j y el (vértice) $_i$ en el que los bordes e_j son los bordes que conectan el (vértice) $_j$ y el (vértice) $_{j+1}$ de dicho segundo polígono y el (vértice) $_i$ es un vértice de dicho primer polígono;
- e. comprobar si dicho primer polígono se encuentra a un lado de L y dicho segundo polígono se encuentra al otro lado de L, y, siempre que el resultado sea negativo;
- f. repetir el paso (e) para todas las combinaciones entre los bordes e_j y los vértices (vértice) $_i$ hasta que se encuentre un plano de separación o hasta que se hayan comprobado todas las combinaciones.

8. Un procedimiento conforme a la reivindicación 3, que comprende:

- a. para cualesquiera n vistas distintas deseadas, generar n listas ordenadas $\{T_{i=1,\dots,n}\}$ de subconjuntos de la malla poligonal que representan cada vista;
- b. generar un gráfico clicable que consiste en n nodos (N_i), siendo dichas n listas T_i ordenadas y de bordes E_{ij} que conectan el nodo i con el nodo $j_{j=1,\dots,n}$;
- c. asignar un peso para cada borde E_{ij} en dicho gráfico clicable que se corresponde con el coste de la diferencia entre dos nodos N_i y N_j cualesquiera;
- d. generar el Árbol de extensión mínima (MST) de dicho gráfico clicable;

ES 2 326 755 T3

- e. representar la totalidad de dichas listas ordenadas $\{T_i\}$ mediante una lista ordenada seleccionada T_k , siendo la raíz de dicho MST, y, por tener todos los bordes el peso más bajo, que formen parte de dicho MST, y yendo desde T_k a todas las listas ordenadas restantes; y
- f. codificar dicha T_k y todas las transiciones representadas mediante dichos bordes seleccionados en un fichero de datos, a partir del que se pueden reconstruir la totalidad de las n listas ordenadas T_i .

5

10

15

20

25

30

35

40

45

50

55

60

65

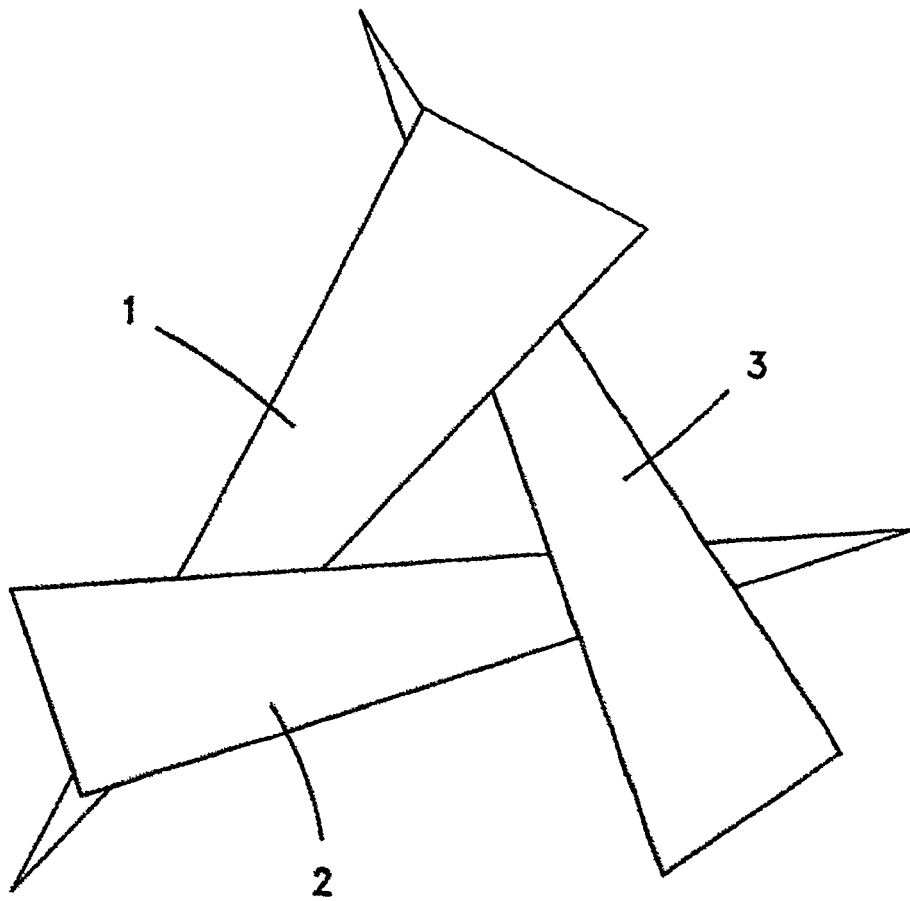


Fig. 1

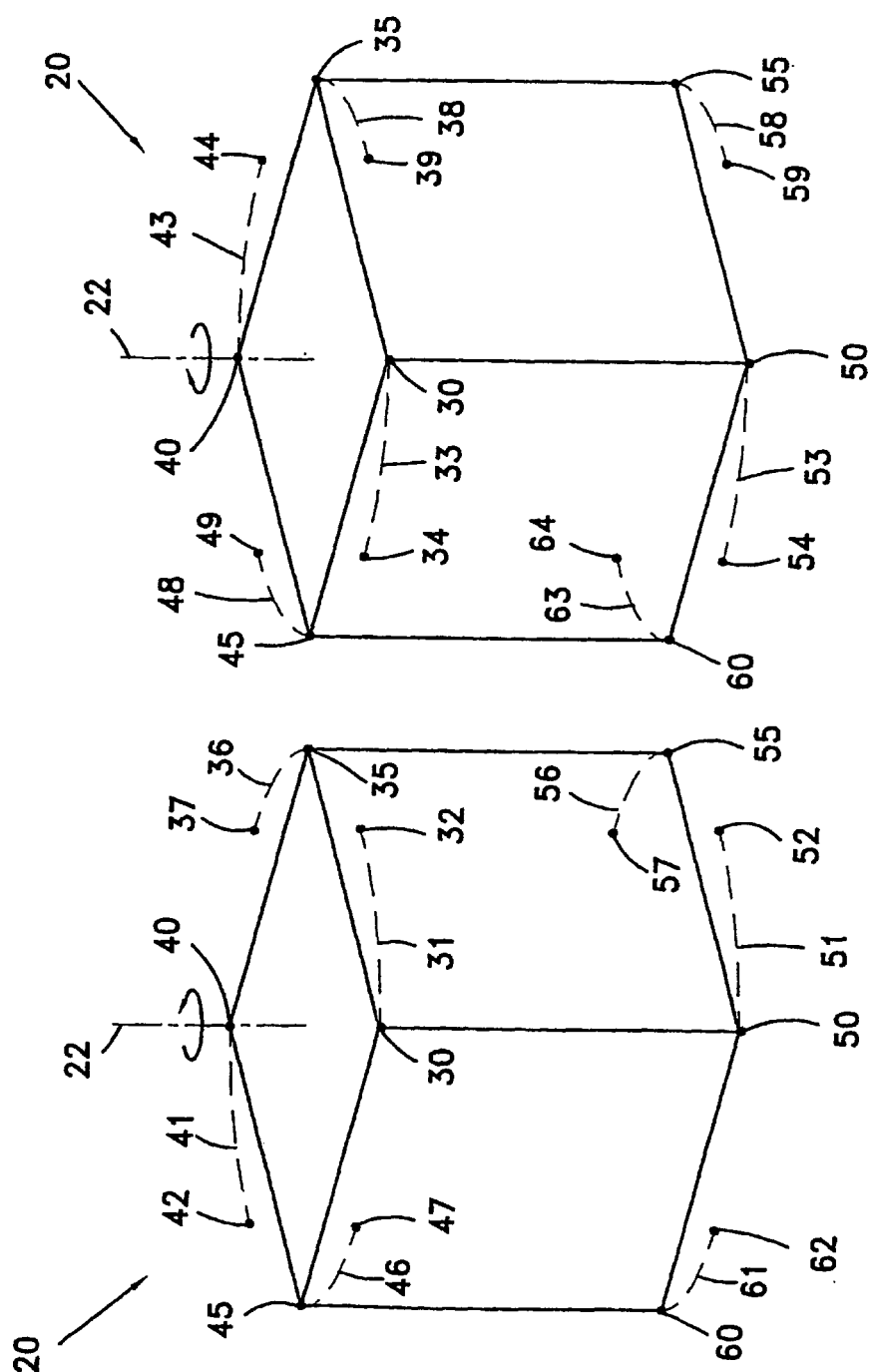
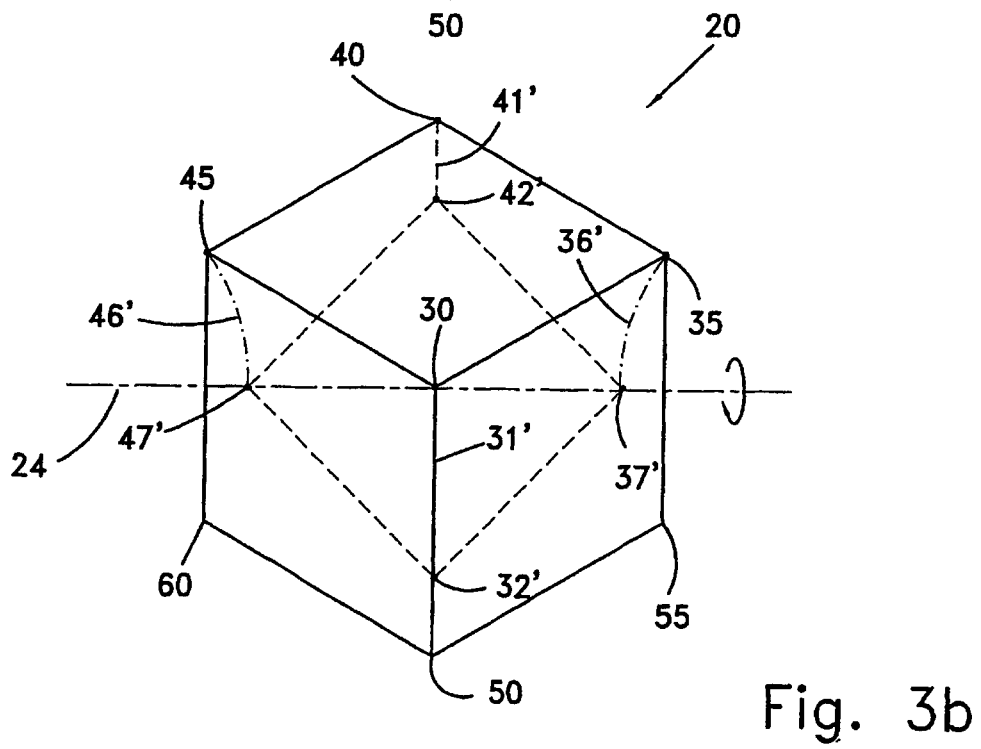
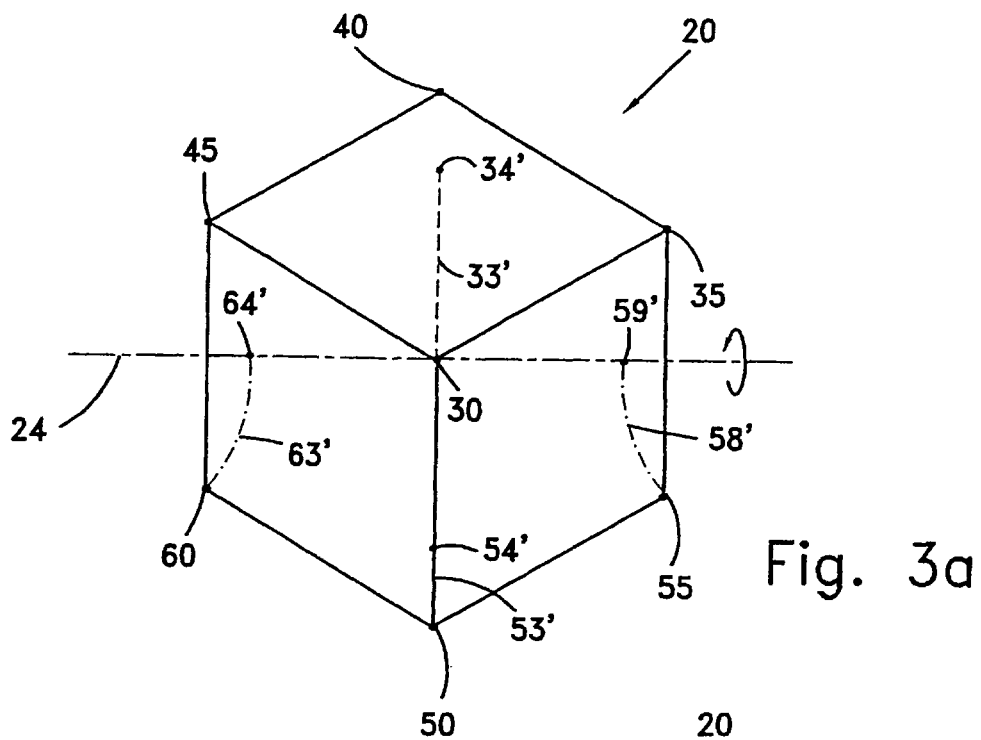


Fig. 2a

Fig. 2b



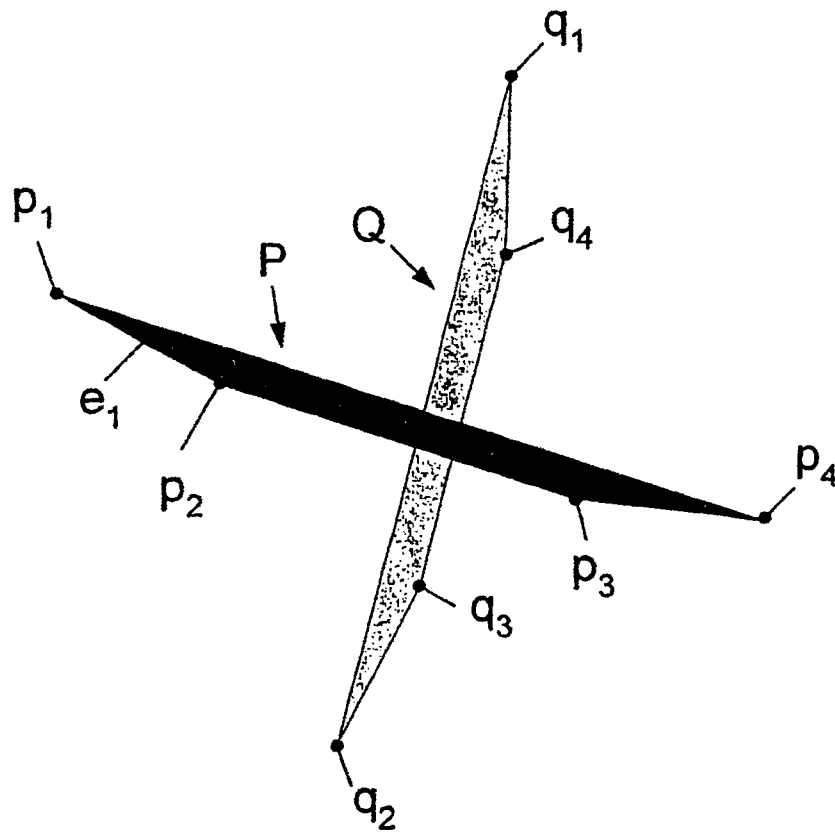


Fig. 4