

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 407 080**

51 Int. Cl.:

**G06T 15/50** (2011.01)

**G06T 11/00** (2006.01)

**G06T 15/20** (2011.01)

**G06T 15/00** (2011.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **23.08.2001 E 01307190 (7)**

97 Fecha y número de publicación de la concesión europea: **19.09.2012 EP 1189173**

54 Título: **Iluminación acromática en un sistema de gráficos y método**

30 Prioridad:

**23.08.2000 US 227007**

**28.11.2000 US 726216**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

**11.06.2013**

73 Titular/es:

**NINTENDO CO., LTD. (100.0%)  
11-1 Kamitoba Hokotate-cho, Minami-ku  
Kyoto 601-8501, JP**

72 Inventor/es:

**DREBIN, ROBERT A.;  
YASUMOTO, YOSHITAKA;  
HOLLIS, MARTIN y  
DEMERS, ERIC**

74 Agente/Representante:

**RIZZO, Sergio**

**ES 2 407 080 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).

## **Iluminación acromática en un sistema de gráficos y método**

### **Descripción**

#### **Campo de la invención**

5 [0001] La presente invención hace referencia a gráficos de ordenador, y más en particular a sistemas de gráficos interactivos como las plataformas domésticas de videojuegos. Más específicamente, esta invención hace referencia a una mejora en el procesamiento de datos que utiliza una función de iluminación para producir un parámetro para su uso posterior al cambiar un color u opacidad. Aún más específicamente, la invención hace referencia a un sistema de gráficos 3D en el que un cálculo de iluminación define un parámetro como la distancia o el ángulo que se aplica a una función adicional (por ejemplo, la texturización).  
10

#### **Antecedentes y resumen de la invención**

15 [0002] La mayoría hemos visto películas que contienen dinosaurios, aliens, juguetes animados y otras criaturas fantásticas extraordinariamente realistas. Dichas animaciones son posibles de hacer gracias a los gráficos por ordenador. Utilizando tales técnicas, un diseñador de gráficos por ordenador puede especificar cómo debería verse cada objeto y como debería cambiar su apariencia a lo largo del tiempo, y un ordenador entonces modela los objetos y los muestra en un visualizador como una televisión o una pantalla de ordenador. El ordenador se encarga de elaborar las variadas tareas requeridas para asegurarse de que cada parte de la imagen mostrada  
20 tiene el color y la forma correcta basándose en la posición y orientación de cada objeto en una escena, la dirección a la que la iluminación parece alcanzar cada objeto, la textura de la superficie de cada objeto, y otros factores.

[0003] Debido a que la generación de gráficos por ordenador es compleja, los gráficos tridimensionales generados por ordenador hace sólo unos años estaban en su mayoría limitados a  
25 caros simuladores de vuelo especializados, terminales de trabajo de alta categoría y superordenadores. El público veía algunas de las imágenes generadas por estos sistemas informáticos en películas y anuncios caros de televisión, pero la mayoría no podía realmente interactuar con los ordenadores que llevaban a cabo la generación de gráficos. Todo esto ha cambiado con la disponibilidad de plataformas de gráficos 3D relativamente económicas, como  
30 por ejemplo, la Nintendo 64® y varias tarjetas gráficas 3D disponibles hoy en día para los ordenadores personales. Ahora es posible interactuar con emocionantes animaciones y simulaciones 3D en sistemas de gráficos informáticos relativamente económicos en casa o en la oficina.

[0004] Lake, Marshall, Harris, Blackstein: "Stylised rendering techniques for scalable real-time 3D  
35 animation" ("Técnicas de representación estilizada para animación 3D escalable a tiempo real") describe métodos de gráficos a tiempo real para emular estilos de dibujos animados. En particular, una ecuación de iluminación se utiliza para calcular la iluminación difusa en los vértices tanto para sombreado suave como para sombreado de dibujos animados.

[0005] Se conoce desde hace tiempo cómo realizar cálculos de iluminación en sistemas de

gráficos 3D basados en una variedad de parámetros (p. ej., atenuación de distancia, ángulo, haz de luz, etc.). Tradicionalmente, los resultados de dichos cálculos de iluminación se utilizaban para modificar el color y/o la opacidad del objeto mostrado - como se documenta en una variedad de materiales estándar de referencia sobre gráficos informáticos (ver por ejemplo, Foley et al, Computer Graphics Principles and Practice (2a. Ed. 1990) en el capítulo 16 ("Illumination and Shading"); y Möller et al, Real Time Rendering (A K Peters 1999) en la sección 4.3 y siguientes ("Lighting and Shading"); Rogers et al, Procedural Elements for Computer Graphics (2a Ed. McGraw-Hill 1997) en la sección 5.2 y siguientes ("Illumination Models"); Neider et al, OpenGL Programming Guide (Addison-Wesley 1993) en el capítulo 6 ("Lighting"); and Kovach, Inside Direct3D (Microsoft Press 2000) at Chapter 5 ("Direct3D Vertices and the Transformation and Lighting Pipeline"). Por ejemplo, las técnicas conocidas como sombreado Gouraud y sombreado Phong modifican el color de una superficie mostrada dependiendo del efecto de luz modelado por una ecuación de iluminación. Como ejemplo, un foco de luz brillante que resplandece sobre una superficie brillante puede tener un efecto de blanqueamiento del color de la superficie en los puntos donde resplandece la luz. Esto se consigue a menudo en un sistema de gráficos convencional con un rasterizador que determina el color de cada píxel en la superficie mostrada de una superficie primitiva basada en el color primitivo (a menudo definido en una base vértice por vértice) y en el resultado de la ecuación o ecuaciones de iluminación.

**[0006]** Un problema al que se enfrentaban los diseñadores de sistemas de gráficos en el pasado era cómo crear de manera eficaz imágenes que no fueran fotorrealistas, como los personajes de dibujos animados. Durante muchos años, la mayoría del trabajo en el campo de los gráficos se centraba en crear imágenes que fueran tan realistas como las fotografías. Sin embargo, más recientemente ha crecido el interés por las imágenes que no son fotorrealistas.

**[0007]** Un tipo de formación de imágenes no fotorrealistas que ha generado interés recientemente es el proceso de automatizar las imágenes de los personajes de dibujos animados. Durante el apogeo de la elaboración a mano de los dibujos animados en los años treinta y cuarenta, los artistas creaban maravillosos personajes de dibujos animados que cambiaban dinámicamente de plano en plano. Estos dibujos animados elaborados a mano de muchas maneras fijan el estándar para la representación de dibujos animados. Los dibujos animados no pretendían parecer realistas. Al contrario, los dibujos animados se diseñaban para aparecer como caricaturas. Por ejemplo, en tales dibujos animados, la cara de una persona puede aparecer blanca como la pared con mejillas rosadas definidas por una coloración roja brillante o un rosa rojizo. A medida que el personaje se mueve a través de la escena, la coloración de las mejillas puede cambiar dinámicamente según coloree a mano el artista cada plano. Dichos efectos dinámicos son fascinantes de ver y añaden interés a las imágenes de los dibujos animados.

**[0008]** Desgraciadamente, el arte de dibujar a mano los dibujos animados de los años treinta y cuarenta era muy caro y llevaba mucho tiempo. Además, la gente ahora quiere utilizar videojuegos y juegos de ordenador para interactuar con los personajes de dibujos animados. Pese a que los videojuegos han tenido éxito durante años en su representación dinámica de los personajes de dibujos animados de manera interactiva, nunca han conseguido los detalles del arte

manual de los últimos dibujos animados dibujados a mano. Pese a que se ha trabajado mucho en el tipo de sistemas de alta categoría para crear dibujos animados y otras imágenes no fotorrealistas, todavía existen posibles y deseables mejoras.

5 **[0009]** US 6.034.693 describe un aparato sintetizador de imagen, y un método sintetizador de imagen, y un medio de almacenamiento para representar el lado frontal y posterior de un polígono. El método de US 6.034.693 determina si debe mostrarse el lado frontal o el posterior de un polígono. Si el lado frontal debe mostrarse, una primera textura se mapea con el polígono, mientras que si se determina que debe mostrarse el lado posterior del polígono una segunda textura se mapea con el polígono.

10 **[0010]** La presente invención proporciona mejoras en el efecto no fotorrealista y otros efectos de imagen que pueden implementarse utilizando un sistema de gráficos de bajo coste como, por ejemplo, una plataforma doméstica de videojuegos o el acelerador de gráficos de un ordenador personal.

15 **[0011]** De acuerdo con la presente invención, se proporciona un método para generar imágenes gráficas en un sistema de generación de gráficos del tipo que define al menos una función de iluminación por vértice; el método comprende el uso de la función de iluminación para calcular al menos un parámetro acromático a utilizar en una modificación subsecuente de un color u opacidad, y la representación de una imagen basada, al menos en parte, en el parámetro calculado, en el que dicha función de iluminación calcula un parámetro negativo para la  
20 retroiluminación.

**[0012]** La función de iluminación es del tipo típicamente utilizado para iluminar objetos dentro de una escena y se utiliza para producir un parámetro diferente al color. Tal parámetro se utiliza para modificar un color u opacidad de un objeto.

25 **[0013]** Un cálculo de iluminación lleva a cabo una iluminación por vértice para proporcionar salidas de componente de color convencionales. Las salidas de componente de color pueden aplicarse a una función de texturización que procesa los componentes de color como parámetros acromáticos. La función de texturización puede proporcionar un efecto de visualización (p. ej. color y/u opacidad modificada basándose en los parámetros acromáticos) que se utiliza para contribuir a la visualización de una escena representada.

30 **[0014]** La salida de cálculo de iluminación puede proporcionar tres componentes de color uno de los cuales se descarta. Los otros dos componentes de color pueden convertirse en coordenadas de textura y utilizarse en una operación de mapeado de textura. Una de las coordenadas de textura puede elegir un mapa de textura unidimensional dentro de una tabla bidimensional. La otra coordenada de textura puede seleccionar un texel particular dentro del mapa de textura  
35 unidimensional seleccionado. El texel resultante proporciona información de color y/o de alfa que se aplica a una superficie definida por el vértice.

**[0015]** Los efectos de iluminación de dibujos animados no fotorrealistas pueden proporcionarse en un sistema de gráficos 3D a tiempo real utilizando un cálculo de iluminación para producir un parámetro diferente al color o a la opacidad. Un mapa de textura unidimensional puede definirse  
40 especificando diferentes pinceladas como una función del parámetro. Una operación convencional

de mapeado de textura puede utilizarse para mapear el mapa de textura unidimensional con una superficie de polígono basada en el parámetro.

**[0016]** Más detalladamente, el sistema publicado puede proporcionar un sombreado convencional basado en la iluminación (como se ha mencionado antes) como parte de su operación global, pero proporciona una mejora adicional que permite que la ecuación o las ecuaciones de iluminación generen una coordenada de textura (es decir, un parámetro diferente a color u opacidad). Ese otro parámetro puede conectarse con otra función adicional (por ejemplo, una tabla de búsqueda registrada como una textura) para proporcionar un resultado adicional - que puede a su vez cambiar el color/opacidad de un objeto. El paso intermedio de hacer que el cálculo de iluminación produzca un parámetro que sólo afecte indirectamente el color y/o la opacidad de un objeto mostrado, proporciona una flexibilidad adicional más allá de los enfoques convencionales que utilizan la salida de ecuación de iluminación para afectar directamente al color de superficie primitivo.

**[0017]** Un ejemplo particular de un modo de realización utiliza un resultado de la ecuación de iluminación para seleccionar una textura unidimensional de un número de texturas. Como ejemplo, para mostrar un tronco de árbol se pueden definir dos texturas unidimensionales diferentes, una con una gama de plateados y la otra con una gama de marrones. La ecuación de iluminación podría utilizarse para determinar el ángulo al que la fuente de iluminación alcanza el tronco de árbol, con el resultado seleccionando entre las dos texturas. El tronco de árbol podría colorearse (utilizando mapeado de textura convencional) con una gama de diferentes plateados desde un ángulo oblicuo pero con una gama de marrones/negros cuando la luz le alcanza de frente. En tal caso, la ecuación de iluminación selecciona una textura unidimensional (es decir, actúa como un parámetro diferente a color u opacidad - en este ejemplo particular, seleccionando dos texturas diferentes).

**[0018]** En el modo de realización particular aquí publicado, se realizan mejoras en la tubería de transformación e iluminación ("T&L") para permitir que genere una coordenada o coordenadas de textura basada/s en la ecuación de iluminación. La transformación e iluminación desarrollan la salida de ecuación de iluminación, pero en vez de proporcionarla directamente para cambiar el píxel de color/opacidad generado por el rasterizador, proporciona el resultado como una coordenada de textura para una unidad de textura. En el modo de realización publicado, la salida de la unidad de textura es un color u opacidad que se mezcla con el color/opacidad primitivo/a del objeto (p. ej., como se desarrolla con la tubería de transformación e iluminación y el rasterizador de un modo convencional basado en la misma ecuación de iluminación o en una diferente utilizando un algoritmo de sombreado convencional como un sombreado de Gouraud o de Phong) para proporcionar un valor de color/opacidad modificado para su visualización.

#### **Breve descripción de los dibujos**

**[0019]** Estas y otras características y ventajas proporcionadas por la invención se entenderán mejor y de manera más completa con las referencias a la siguiente descripción detallada de los presentes modos de realización preferidos junto con los dibujos en los que:

la figura 1 es una vista global de un ejemplo de sistema interactivo de gráficos por ordenador;

la figura 2 es un diagrama de bloques del ejemplo de sistema de gráficos por ordenador de la figura 1;

la figura 3 es un diagrama de bloques del ejemplo de procesador de gráficos y de audio mostrado en la figura 2;

la figura 4 es un diagrama de bloques del ejemplo de procesador de gráficos 3D mostrado en la figura 3;

la figura 5 es un ejemplo de diagrama de flujo lógico del procesador de gráficos y de audio de la figura 4;

la figura 6 muestra un ejemplo de proceso de generación de imagen de alto nivel incluido en esta invención;

la figura 7 muestra un proceso de generación de imagen más específico incluido en esta invención que utiliza texturización basada en un parámetro de iluminación acromático;

la figura 8A muestra un ejemplo de función de textura;

la figura 8B muestra un ejemplo de mapa de textura unidimensional;

la figura 9 muestra un ejemplo de proceso de mapeado de textura;

la figura 10A muestra un ejemplo de mapa de textura bidimensional;

la figura 10B muestra un ejemplo de mapa de textura bidimensional;

la figura 11 muestra un ejemplo de implementación de la tubería de iluminación;

la figura 12 muestra un ejemplo de asociación de luces con canales de color;

la figura 13 muestra un ejemplo de implementación de unidad de transformación;

la figura 14 muestra un ejemplo de implementación de tubería de iluminación;

la figura 15 muestra un ejemplo de implementación de la unidad de multiplicación/suma de vector;

la figura 16 muestra un ejemplo de implementación del normalizador;

la figura 17 muestra un ejemplo de implementación de la distancia de atenuación;

la figura 18 muestra un ejemplo de implementación de la escala de iluminación;

la figura 19 muestra un ejemplo de implementación del acumulador de enteros;

la figura 20 muestra un ejemplo de diagrama de transición de estado; y

las figuras 21A y 21B muestran ejemplos de implementaciones alternativas compatibles.

### **Descripción detallada de los ejemplos de modos de realización de la invención**

**[0020]** La figura 1 muestra un ejemplo de sistema interactivo de gráficos 3D por ordenador 50. El sistema 50 puede utilizarse para reproducir videojuegos interactivos en 3D con un sonido estéreo interesante. También puede utilizarse para una variedad de diferentes aplicaciones.

**[0021]** En este ejemplo, el sistema 50 es capaz de procesar, de manera interactiva a tiempo real, una representación digital o un modelo de un mundo tridimensional. El sistema 50 puede mostrar una parte o la totalidad del mundo desde un punto de vista arbitrario. Por ejemplo, el sistema 50 puede cambiar de manera interactiva el punto de vista en respuesta a las entradas a tiempo real

de los controladores portátiles 52a, 52b u otros dispositivos de entrada. Esto permite al jugador ver el mundo a través de los ojos de alguien dentro o fuera del mundo. El sistema 50 puede utilizarse para aplicaciones que no requieran una reproducción interactiva 3D a tiempo real (p. ej., un visualizador que genere 2D y/o un visualizador que no sea interactivo), pero la capacidad de generar imágenes 3D de calidad de forma muy rápida puede utilizarse para crear una partida muy realista y emocionante u otras interacciones gráficas.

**[0022]** Para jugar a un videojuego u otra aplicación utilizando el sistema 50, el usuario primero conecta una unidad principal 54 a su televisión a color 56 u otro dispositivo al conectar un cable 58 entre ambos. La unidad principal 54 produce tanto señales de vídeo como de audio para controlar la televisión a color 56. Las señales de vídeo son las que controlan las imágenes mostradas en la pantalla de la televisión 59, y las señales de audio se reproducen como sonido a través de los altavoces estéreo de la televisión 61L, 61R.

**[0023]** El usuario también necesita conectar la unidad principal 54 a la fuente de energía. Esta fuente de energía puede ser un adaptador de corriente alterna convencional (no mostrado) que se conecta a un enchufe eléctrico doméstico estándar y convierte la corriente doméstica en una señal de voltaje CC adecuada para suministrar energía a la unidad principal 54. Pueden utilizarse baterías en otros modos de realización.

**[0024]** El usuario puede utilizar controladores manuales 52a, 52b para controlar la unidad principal 54. Los controles 60 pueden utilizarse, por ejemplo, para especificar la dirección (arriba o abajo, izquierda o derecha, más cerca o más lejos) a la que el personaje mostrado en la televisión 56 debería moverse en un mundo 3D. Los controles 60 también proporcionan una entrada a otras aplicaciones (p. ej., selección del menú, control del puntero/cursor, etc.). Los controladores 52 pueden tomar diversas formas. En este ejemplo, los controladores 52 mostrados incluyen cada uno controles 60 como joysticks, pulsadores de botón y/o interruptores de dirección. Los controladores 52 pueden conectarse a la unidad principal 54 mediante cables o sin ellos mediante ondas electromagnéticas (por ejemplo, de radio o infrarrojas).

**[0025]** Para reproducir una aplicación como un juego, el usuario selecciona un medio de almacenamiento apropiado 62 grabando el videojuego u otra aplicación a la que quiera jugar, e inserta ese medio de almacenamiento en una ranura 64 en la unidad principal 54. El medio de almacenamiento 62 puede, por ejemplo, ser un disco óptico y/o magnético especialmente codificado y/o encriptado. El usuario puede manejar un interruptor de energía 66 para encender la unidad principal 54 y provocar que la unidad principal comience a reproducir el videojuego y otra aplicación basada en el software grabado en el medio de almacenamiento 62. El usuario puede utilizar los controladores 52 para suministrar entradas a la unidad principal 54. Por ejemplo, el uso de un control 60 puede iniciar el juego u otra aplicación. Mover otros controles 60 puede provocar que los personajes animados se muevan en direcciones diferentes o que cambie el punto de vista del usuario en un mundo 3D. Dependiendo del software particular almacenado en el medio de almacenamiento 62, los diferentes controles 60 en el controlador 52 pueden llevar a cabo diferentes funciones en diferentes momentos.

**Ejemplo electrónico de todo el sistema.**

**[0026]** La figura 2 muestra un diagrama de bloques de ejemplos de componentes del sistema 50. Los componentes primarios incluyen:

- un procesador principal (CPU) 110,
- una memoria principal 112, y
- un procesador de gráficos y de audio 114.

**[0027]** En este ejemplo, el procesador principal 110 (p. ej., un PC IBM Power 750 aumentado) recibe entradas de controladores portátiles 108 (y/u otros dispositivos de entrada) a través de un procesador de audio y de gráficos 114. El procesador principal 110 responde de manera interactiva a las entradas del usuario, y ejecuta un videojuego u otro programa suministrado, por ejemplo, por un medio de almacenamiento externo 62 a través de un dispositivo de acceso al almacenamiento masivo 106 como una unidad de disco óptico. Como un ejemplo, en el contexto de los videojuegos, el procesador principal 110 puede incluir la detección de colisión y el procesamiento de animación además de una variedad de funciones interactivas y de control.

**[0028]** En este ejemplo, el procesador principal 110 genera gráficos 3D y órdenes de audio y las envía al procesador de gráficos y audio 114. El procesador de gráficos y de audio 114 procesa estas órdenes para generar imágenes visuales interesantes en el visualizador 59 y sonido estéreo interesante en los altavoces estéreo 61R, 61L u otros dispositivos adecuados para generar sonido.

**[0029]** El sistema de ejemplo 50 incluye un codificador de vídeo 120 que recibe señales de imagen de un procesador de gráficos y audio 114 y convierte las señales de imagen en señales de vídeo analógicas y/o digitales adecuadas para mostrar en un dispositivo visualizador estándar como un monitor de ordenador o una televisión doméstica a color 56. El sistema 50 también incluye un códec de audio (codificador/decodificador) 122 que codifica y decodifica señales de audio digitalizadas y también puede convertir formatos de señales de audio digitales y analógicas cuando sea necesario. El códec de audio 122 puede recibir entradas de audio a través de una memoria 124 y enviarlas al procesador de gráficos y audio 114 para su procesamiento (es decir, al mezclarlas con otras señales de audio, el procesador genera y/o recibe una salida a través de una transferencia de audio del dispositivo de acceso de almacenamiento masivo 106). El procesador de gráficos y audio 114 en este ejemplo puede almacenar información relacionada con el audio en una memoria de audio (SDRAM) 126 que está disponible para tareas de audio. El procesador de gráficos y audio 114 proporciona las señales de salida de audio resultantes al códec de audio 122 para su decodificación y conversión en señales analógicas (por ejemplo, a través de amplificadores de memoria 128L, 128R) para que puedan reproducirse en los altavoces 61L, 61R.

**[0030]** El procesador de gráficos y audio 114 tiene la habilidad de comunicarse con varios dispositivos adicionales que pueden estar presentes en el sistema 50. Por ejemplo, un bus paralelo digital 130 puede utilizarse para comunicar con un dispositivo de acceso de almacenamiento masivo 106 y/u otros componentes. Un bus serial periférico 132 puede comunicar con una variedad de dispositivos periféricos o de otro tipo incluyendo, por ejemplo:

- una memoria programable de sólo lectura (PROM) y/o un reloj de tiempo real (RTC) 134,
- un módem 136 u otra interfaz de red (que puede a su vez conectar el sistema 50 a una red de

telecomunicaciones 138 como internet u otra red digital desde/a la que pueden descargarse o subirse instrucciones de programa y/o datos), y

- una memoria flash 140.

Un bus serial externo adicional 142 puede utilizarse para comunicar con una memoria de expansión adicional 144 (p. ej., una tarjeta de memoria) u otros dispositivos. Los conectores pueden utilizarse para conectar varios dispositivos a los buses 130, 132, 142.

### **Ejemplo de procesador de gráficos y de audio**

[0031] La figura 3 es un diagrama de bloques de un ejemplo de procesador de gráficos y de audio 114. El procesador de gráficos y de audio 114 en un ejemplo puede ser un ASIC (Circuito Integrado para Aplicaciones Específicas) de chip único. En este ejemplo, el procesador de gráficos y audio 114 incluye:

- una interfaz de procesador 150,
- una interfaz o un controlador de memoria 152,
- 15 • un procesador de gráficos 3D 154,
- un procesador digital de señales de audio (DSP) 156,
- una interfaz de memoria de audio 158,
- una interfaz y un mezclador de audio 160,
- un controlador periférico 162, y
- 20 • un controlador de visualización 164.

[0032] El procesador de gráficos 3D 154 lleva a cabo tareas de procesamiento de gráficos. El procesador digital de señales de audio 156 lleva a cabo tareas de procesamiento de audio. El controlador de visualización 164 accede a la información de imagen desde la memoria principal 112 y la envía al codificador de vídeo 120 para su visualización en el dispositivo de visualización 25 56. La interfaz de audio y el mezclador 160 interactúa con el códec de audio 122, y también pueden mezclar audio de diferentes fuentes (p. ej., la transferencia de audio desde el dispositivo de acceso de almacenamiento masivo 106, la salida de audio del DSP 156, y la entrada de audio externa recibida a través del códec 122). La interfaz del procesador 150 proporciona una conexión de datos y de control entre el procesador principal 110 y el procesador de gráficos y audio 114.

[0033] La interfaz de memoria 152 proporciona una conexión de datos y de control entre el procesador de gráficos y de audio 114 y la memoria 112. En este ejemplo, el procesador principal 110 accede a la memoria principal 112 a través de la interfaz del procesador 150 y de la interfaz de memoria 152 que son parte del procesador de gráficos y de audio 114. El controlador periférico 162 proporciona una conexión de datos y de control entre el procesador de gráficos y de audio 35 114 y los varios periféricos arriba mencionados. La interfaz de memoria de audio 158 proporciona una conexión con la memoria de audio 126.

### **Ejemplo de tubería de gráficos**

[0034] La figura 4 muestra una vista más detallada de un ejemplo del procesador de gráficos 3D

154. El procesador de gráficos 3D 154 incluye, entre otras cosas, un procesador de órdenes 200 y una tubería de gráficos 3D 180. El procesador principal 110 comunica corrientes de datos (p. ej., corrientes de órdenes de gráficos y listas de visualización) al procesador de órdenes 200. El procesador principal 110 tiene dos niveles de caché 115 para minimizar la latencia de memoria, y también tiene un búfer o memoria de escritura-acumulación 111 para las corrientes de datos no guardadas en la caché destinadas al procesador de gráficos y de audio 114. El búfer de escritura-acumulación recoge líneas parciales de caché en líneas completas de caché y envía los datos al procesador de gráficos y audio 114 línea por línea para conseguir el máximo uso del bus.

**[0035]** El procesador de órdenes 200 recibe órdenes de visualización desde el procesador principal 110 y las analiza - obteniendo cualquier dato adicional necesario para procesarlas desde la memoria compartida 112. El procesador de órdenes 200 proporciona una corriente de órdenes de vértice a la tubería de gráficos 180 para el procesamiento y la representación 2D y/o 3D. La tubería de gráficos 180 genera imágenes basadas en estas órdenes. La información de imagen resultante puede transferirse a la memoria principal 112 para acceder mediante la unidad de interfaz del controlador de visualización o vídeo 164 - que muestra la salida de la memoria de pantalla (*framebuffer*) de la tubería 180 en el visualizador 56.

**[0036]** La figura 5 es un diagrama de flujo lógico del procesador de gráficos 154. El procesador principal 110 puede almacenar corrientes de órdenes gráficas 210, listas de visualización 212 y selecciones de vértices 214 en la memoria principal 112, y pasar indicaciones al procesador de órdenes 200 a través de la interfaz de bus 150. El procesador principal 110 almacena órdenes de gráficos en uno o más búferes de gráficos de primero en entrar, primero en salir (*FIFO*) 210 que reparte en la memoria principal 110. El procesador de órdenes 200 extrae:

- corrientes de órdenes desde la memoria principal 112 a través de un búfer de memoria *FIFO* en chip 216 que recibe y registra las órdenes gráficas para el control de la sincronización o el flujo y el equilibrio de la carga,
- listas de visualización 212 de la memoria principal 112 a través de una llamada del búfer de memoria *FIFO* en chip 218, y
- atributos de vértices desde la corriente de órdenes y/o desde las selecciones de vértices 214 en la memoria principal 112 a través de una caché de vértice 220.

**[0037]** El procesador de órdenes 200 lleva a cabo órdenes de operaciones de procesamiento 200a que convierten tipos de atributos en formato de punto flotante, y pasan los datos resultantes del vértice del polígono a la tubería de gráficos 180 para su representación/rasterización. Una circuitería de arbitraje de la memoria programable 130 (ver figura 4) arbitra el acceso a la memoria principal compartida 112 entre el tubería de gráficos 180, el procesador de órdenes 200 y la unidad de interfaz del controlador de visualización o vídeo 164.

**[0038]** La figura 4 muestra que la tubería de gráficos 180 puede incluir:

- una unidad de transformación 300,
- un configurador/rasterizador 400,
- una unidad de textura 500,
- una unidad de entorno de textura 600, y

- un motor de píxel 700.

**[0039]** La unidad de transformación 300 lleva a cabo una variedad de transformaciones 2D y 3D y otras operaciones 300a (ver figura 5). Una unidad de transformación 300 puede incluir una o más memorias de matriz 300b para registrar matrices utilizadas en el proceso de transformación 300a.

5 La unidad de transformación 300 transforma la geometría de entrada por vértice desde un espacio de objeto a un espacio de pantalla; y transforma las coordenadas de textura entrantes y calcula las coordenadas de textura de proyección (300c). La unidad de transformación 300 también puede llevar a cabo un recorte/selección de polígono 300d. El procesamiento de iluminación 300e también elaborado por la unidad de transformación 300b proporciona cálculos de iluminación por  
10 vértice para hasta 8 luces independientes en un ejemplo del modo de realización. La unidad de transformación 300 también lleva a cabo la generación de coordenadas de textura (300c) para efectos de mapeado de rugosidad del tipo alzado, así como operaciones de recorte/selección de polígono (300d).

**[0040]** El configurador/rasterizador 400 incluye una unidad de configuración que recibe datos de vértice desde la unidad de transformación 300 y envía información de la configuración triangular a una o más unidades de rasterización (400b) llevando a cabo rasterización de aristas, la rasterización de coordenadas de textura y la rasterización de color.

**[0041]** La unidad de textura 500 (que puede incluir una memoria de textura en chip (TMEM) 502) lleva a cabo varias tareas relacionadas con la texturización incluyendo por ejemplo:

- recuperar texturas 504 de la memoria principal 112,
- procesar texturas (500a) incluyendo, por ejemplo, el uso de multitexturizado, la descompresión de textura post-caché, el filtrado de textura, el realzado, el sombreado y la iluminación a través del uso de texturas de proyección, y el BLIT con transparencia y profundidad alfa,
- 25 • procesar un mapa de rugosidad para calcular los desplazamientos de coordenadas para el mapeado de rugosidad, pseudo textura y efectos de textura de mosaico (500b), y
- el procesamiento indirecto de textura (500c).

**[0042]** La unidad de textura 500 envía valores de textura filtrados a la unidad de entorno de textura 600 para el procesamiento del entorno de textura (600a). La unidad de entorno de textura  
30 600 mezcla el polígono y el color, o el valor alfa, o la profundidad de la textura, y también puede llevar a cabo el procesamiento de niebla de textura (600b) para conseguir una escala inversa basada en efectos de niebla. La unidad de entorno de textura 600 puede proporcionar múltiples etapas para elaborar una variedad de otras funciones interesantes relacionadas con entornos basadas por ejemplo en la modulación de color/alfa, realzado, texturización detallada, intercambio  
35 de textura, fijación, y mezcla de profundidad.

**[0043]** El motor de píxel 700 realiza una comparación (700a) de profundidad (z) y una mezcla de píxel (700b). En este ejemplo, el motor de píxel 700 guarda los datos en una memoria integrada del tipo *framebuffer* (en chip) (FBi) 702. La tubería de gráficos 180 puede incluir una o más memorias dinámicas de acceso aleatorio (DRAM) incorporadas 702 para guardar información de  
40 la memoria *framebuffer* (FBi) y/o información de textura de manera local. La comparación del

búfer Z 700a también puede llevarse a cabo en una etapa anterior en la tubería de gráficos 180 dependiendo del modo de representación en efecto en la actualidad (p. ej., las comparaciones z pueden elaborarse antes si la mezcla alfa no es necesaria). El motor de píxel 700 incluye una operación de copia 700c que periódicamente escribe en la memoria *framebuffer* en chip 702 sobre la memoria principal 112 para acceder mediante la unidad de interfaz de visualización/vídeo 164. Esta operación de copia 700c también puede utilizarse para copiar los contenidos de la memoria integrada *framebuffer* 702 a las texturas en la memoria principal 112 para conseguir efectos de síntesis de textura dinámica. El antialiasing y otros filtrados pueden llevarse a cabo durante la operación de copia. La salida de la memoria *framebuffer* de la tubería de gráficos 180 (que se guarda finalmente en la memoria principal 112) es leída plano a plano por la unidad de interfaz del visualizador/vídeo 164. La interfaz del controlador de visualización/vídeo 164 proporciona valores de píxel RGB digital para visualizar en el visualizador 102.

#### **Ejemplo de función de iluminación acromática**

**[0044]** Como se ha mencionado con anterioridad, la unidad de transformación 300 en el modo de realización de ejemplo elabora transformaciones de iluminación así como geométricas, de recorte, selección y otras funciones. En el modo de realización de ejemplo, la unidad de transformación 300 admite la iluminación en hardware como un cálculo por vértice. Esto significa que un valor de color (RGB) puede calcularse para cada vértice iluminado, y que estos colores pueden estar interpolados de manera lineal sobre la superficie de cada triángulo iluminado. Esto se conoce como el sombreado Gouraud. A continuación se analizan detalles adicionales que conciernen un ejemplo de implementación de iluminación en conexión con las figuras 12 y siguientes.

**[0045]** El ejemplo del modo de realización de la unidad de transformación 300 proporciona un uso adicional para la función o funciones de iluminación 302 no disponible en los sistemas gráficos tradicionales. Generalmente, la salida de la función de iluminación en el sistema de gráficos tradicional es un color utilizado para el sombreado de polígono. Algunos sistemas gráficos proporcionan un valor alfa (p. ej., transparencia) basado en una función de iluminación. En el ejemplo de modo de realización proporcionado por la presente invención, se utiliza una función de iluminación para proporcionar un parámetro acromático (es decir, ni de color ni alfa, sino algo diferente) utilizado para modificar posteriormente un color y/u opacidad.

**[0046]** La figura 6 es un diagrama de bloques que muestra tal configuración cuando la función de iluminación 302 proporciona un parámetro acromático utilizado en un proceso posterior 304 para modificar un color u opacidad. Los resultados de la modificación se muestran en un dispositivo de visualización como un visualizador 56. En este contexto, el término "función de iluminación" se refiere generalmente a una función matemática de una posición de superficie, una orientación de superficie, una ubicación del ojo y, en algunos casos, una ubicación de luz. Las funciones de iluminación pueden definirse como cualquier combinación de estas coordenadas.

**[0047]** En el modo de realización de ejemplo, el parámetro acromático extraído de la función de iluminación 302 se procesa mediante una función de texturización 306 como se muestra en la Figura 7. A un nivel muy simple, la texturización de un objeto significa "pegar" una imagen a ese

objeto. Con más detalle, la texturización generalmente funciona modificando los valores usados en la función de iluminación 302. Por ejemplo, en el caso típico, una función de iluminación 302 se utiliza para determinar el color y la iluminación de una superficie de polígono, y una función de texturización se utiliza para modificar ese color/iluminación para proporcionar un efecto adicional (p. ej., para crear la apariencia de una pared de ladrillos o una veta de madera, para hacer que la superficie parezca abultada, reflejada o envuelta, etc.). En el ejemplo de modo de realización mostrado en la figura 7, en contraste, la salida de la función de iluminación 302 no es un valor de color o de transparencia, sino un parámetro generalizado (por ejemplo, una coordenada de textura) para la función de texturización 306. La función de texturización desarrolla información de color y/u opacidad utilizada para modificar (por ejemplo, mezclar con) un color y/u opacidad primitivos para visualizar en un dispositivo de visualización 56.

**[0048]** La figura 8A muestra un ejemplo de función de textura 306. En este ejemplo, la función de iluminación 302 produce un parámetro S de salida acromática que proporciona parámetros a la función de textura 306 de ejemplo. En este ejemplo de la figura 8A, la función de textura 306 de ejemplo selecciona uno de los tres colores (C1, C2, C3) u opacidades dependiendo del valor del parámetro S. Así, el color u opacidad particular extraído/a de la función de textura 306 se determina como una función del parámetro S extraída de la función de iluminación 302. Debido a que la función de iluminación 302 proporciona el parámetro S, la selección de su color u opacidad extraído/a de la función de textura 306 puede depender de cualquiera de los diferentes factores utilizados por el cálculo de la función de iluminación incluyendo por ejemplo:

- distancia,
- atenuación,
- ángulo,
- vector,
- coseno,
- polinomio,
- tipo de fuente de luz,
- otro coeficiente.

**[0049]** A la vez que la función de textura 306 puede implementarse en una variedad de diferentes formas, un enfoque económicamente eficaz consiste en almacenar los valores de salida de la función de textura 306 en una tabla de búsqueda o mapa de textura y acceder a esos valores utilizando coordenadas de textura. La figura 8B muestra un ejemplo de mapa de textura unidimensional 308 indexado mediante el parámetro S extraído de la función de iluminación 302. Una ventaja al utilizar una función de mapeado de textura convencional en el modo de realización de ejemplo es que el hardware de mapeado de textura u otra funcionalidad disponible para un uso general también puede utilizarse para implementar una función de textura 306 en colaboración con la función de iluminación 302.

**[0050]** La figura 9 muestra un ejemplo más detallado incluyendo una operación de mapeado de textura bidimensional 306. En este ejemplo, un cálculo de iluminación 302 se lleva a cabo basado

en una identificación de vértices y definiciones de iluminación. La salida del cálculo de iluminación 302 en el modo de realización de ejemplo comprende colores por vértice (p. ej., rojo, verde, azul) y/o transparencia (es decir, alfa). El modo de realización de ejemplo utiliza típicamente estas salidas de color del cálculo de iluminación 302 para el sombreado Gouraud de una superficie de polígono. Sin embargo, en este modo de operación, uno de los canales de color (p. ej., azul) se desecha, y los otros dos canales de color (es decir, rojo, verde) se convierten en coordenadas de textura  $s, t$  utilizando una operación de conversión 308. Las coordenadas de textura  $s, t$  se aplican a una o más operaciones de mapeado de textura 306a, 306b, etc. para extraer texels de color/transparencia. La operación de mezclado 602 puede mezclar estos texels con la información del color de la superficie (que puede deducirse utilizando técnicas de sombreado convencionales basadas en el mismo cálculo de iluminación o en uno diferente) para proporcionar polígonos sombreados y texturizados para su visualización.

**[0051]** En el modo de realización de ejemplo, la operación de mapeado de textura 306 proporciona un mapeado de textura bidimensional mostrado en la figura 10A. En este ejemplo, la coordenada de textura  $t$  se genera basándose en la salida del canal de color verde del cálculo de iluminación 302, y se utiliza para seleccionar uno de los mapas de textura unidimensional  $n$  dentro de un mapa de textura bidimensional 310. La otra coordenada de textura  $s$  se desarrolla basándose en la salida de canal de color rojo del cálculo de iluminación 302 y se utiliza para seleccionar un texel en particular dentro de una textura unidimensional seleccionada por la coordenada  $t$ . Por supuesto, la coordenada de textura  $t$  podría seleccionar el mapa unidimensional (1D) y la coordenada de textura  $s$  podría seleccionar un texel en particular en otras implementaciones, y/o podrían utilizarse diferentes colores y/u opacidades para generar las diferentes coordenadas de textura. Tres coordenadas de textura podrían utilizarse para seleccionar texels en una textura 3D si se desea.

**[0052]** La figura 10B es un ejemplo ilustrativo de un mapa de textura bidimensional 310 comprendiendo un número de mapas de textura unidimensional 308. El mapa de textura bidimensional en particular mostrado en la figura 10B es particularmente adecuado para efectos de iluminación de dibujos animados. El ejemplo mostrado incluye un número de mapas de textura unidimensional incluyendo cuatro mapas 308(1) ... 308(4) comprendiendo cada uno dos tipos diferentes de texels (por ejemplo texels morados y texels azules). Una de las coordenadas de textura (p. ej.,  $t$ ) se selecciona de entre estos diferentes mapas de textura y la otra coordenada de textura (p.ej.,  $s$ ) se selecciona de un texel en particular (es decir, azul o morado). Los diferentes mapas de textura unidimensionales 308(1)... 308(4) proporcionan diferentes mapeados entre las coordenadas de textura y los texels azules o morados para proporcionar diferentes pinceladas u otros efectos.

**[0053]** Como otro ejemplo, el mapa de textura unidimensional 308(6) mostrado en la figura 10B incluye cuatro tipos diferentes de texels (amarillo, naranja, rojo y marrón). Estos colores diferentes podrían utilizarse para conseguir llamativos efectos de iluminación del tipo de los dibujos animados donde, por ejemplo, el ángulo que hace una iluminación direccional con un objeto o la distancia de un objeto desde la fuente de iluminación determina el color resultante de la salida de

la operación de mapeado de textura 306. Dicha visualización puede tener una variedad de aplicaciones interesantes - especialmente en una representación en tiempo real no fotorealista como la animación de un dibujo generada de forma dinámica. Por ejemplo, es posible definir la iluminación virtual de un dibujo animado que ilumina un objeto. La visualización representada del objeto depende de la posición del vértice, la posición de la iluminación local del dibujo animado y otros factores que el cálculo de iluminación 302 toma en cuenta (p. ej., el cálculo difuso o especular, la distancia de atenuación, etc.). En un ejemplo, es posible establecer el canal del color rojo en cero y permitir el cálculo de iluminación 302 para calcular el canal verde para especificar un valor de texel en particular dentro de una textura unidimensional (1D) que define un conjunto de pinceladas. El procesador principal de color del material 110 típicamente puede aplicarse al cálculo de iluminación 302, puede utilizarse para especificar qué textura 1D seleccionar dentro del mapa de textura 2D. El cálculo de iluminación 302 calcula qué pincelada seleccionar en una base vértice por vértice sin un coste extra más allá del suministro de un índice por vértice.

**[0054]** A la vez que el ejemplo mostrado en la figura 9 es particularmente útil para la iluminación de dibujos animados, también puede utilizarse para muchas otras aplicaciones. Como por ejemplo, la función de iluminación 302 puede utilizarse para calcular un volumen o superficie de sombra. El cálculo de iluminación 302 basado en un vértice dado podría generar un volumen de sombra definido en coordenadas de textura utilizadas para una operación de mapeado de textura 306. Son posibles muchas otras aplicaciones. Por ejemplo, una aplicación interesante hace referencia a texturas de proyección. Una propiedad no deseable de las texturas de proyección es que típicamente se proyectan hacia delante y hacia atrás desde la cámara. Las técnicas aquí publicadas pueden generar coordenadas de texturas de proyección basándose en una función de iluminación que proporciona atenuación (p. ej., para eliminar la iluminación orientada hacia atrás) y/o la distancia de atenuación (p. ej., para atenuar la textura de proyección basándose en la longitud a la que se proyecta; tal y como se vería al utilizar un proyector de diapositivas real para proyectar una imagen). Dicha texturización de proyección combinada con la función de iluminación acromática aquí descrita puede producir muchas imágenes y efectos de imagen interesantes.

**[0055]** Una operación de conversión 306 como se muestra en la figura 9 puede no ser necesaria en todas las implementaciones. En el modo de realización de ejemplo, el cálculo de iluminación 302 genera una salida íntegra que el bloque de conversión 308 convierte en una representación del punto flotante adecuada para las coordenadas de textura utilizadas en la siguiente operación de mapeado de textura. Sin embargo, dicha conversión de entero a punto flotante podría no ser necesaria en otras implementaciones, u otras implementaciones podrían utilizar diferentes tipos de conversiones, según se desee.

**[0056]** En el modo de realización de ejemplo mostrado en la figura 9, el cálculo de iluminación 302 no necesita calcular las dos coordenadas de textura  $s$  y  $t$ . Ya que sólo una coordenada de textura se utiliza como un parámetro en el modo de realización de ejemplo para seleccionar qué texel en particular en un mapa de textura unidimensional debería aplicarse a una superficie primitiva, el modo de realización preferido calcula preferiblemente esa coordenada de textura (o un valor

intermedio del cual se deduce la coordenada de textura) utilizando un cálculo de iluminación 302. Sin embargo, dependiendo de la aplicación, podría ser deseable para el programa de aplicación, al contrario que para el cálculo de iluminación 302, especificar otra coordenada de textura utilizada para elegir (en este ejemplo de modo de realización) entre diversas texturas unidimensionales 308. Como ejemplo, el programa de aplicación en funcionamiento en el procesador principal 110 puede especificar un valor para cualquiera o todas las salidas del canal de color del cálculo de iluminación 302 a través de definiciones de iluminación aplicadas a la entrada del cálculo de iluminación. En algunas aplicaciones, puede ser deseable para el procesador principal 110 especificar los contenidos de, por ejemplo, el canal de color verde y por consiguiente la textura unidimensional particular usada en la operación de mapeado de textura 306. Esto proporciona un control adicional al programador de aplicación a la vez que permite al cálculo de iluminación 302 generar de forma dinámica la coordenada de textura s basada en las variables que la unidad de cálculo de iluminación 302 evalúa. Los valores del punto flotante que el procesador principal 110 envía a la unidad de transformación pueden truncarse a RGB8 para permitir que los cálculos se elaboren con mayor precisión.

**[0057]** En el modo de realización de ejemplo, no hay razón para que el cálculo de iluminación 302 no pueda producir una salida de valor negativo. En el ejemplo mostrado en la figura 9, esto corresponde a la retroiluminación de un objeto. Por lo tanto, es posible tener una iluminación negativa o retroiluminación de dibujo animado en el modo de realización de ejemplo. Esto podría ser útil para proporcionar efectos donde existe información de la parte posterior de un objeto. El modo de realización preferido soporta un tipo negativo de cálculo de iluminación, pero depende de la aplicación si se añade una escala y/o una polarización para que el valor final acabe siendo positivo. Por ejemplo, la salida del cálculo de iluminación podría mapearse de -1 a 0 y desde +1 a 1 antes de convertirse en una coordenada o coordenadas de textura. El modo de realización de ejemplo no ajusta valores negativos, sino que simplemente los mapea en valores numéricos positivos, siendo la aplicación cuidadosa para asegurarse de que estos valores positivos se interpretan de manera apropiada. Si se desea, otras implementaciones pueden soportar valores de iluminación negativos directamente, por supuesto.

**[0058]** La operación de mapeado de textura 306 y la operación de mezclado 602 mostrada en la figura 9 puede ser arbitrariamente compleja. Por ejemplo, es posible mezclar con alfa. Según otro ejemplo, será posible utilizar dos canales de color separados extraídos del cálculo de iluminación 302 para dos operaciones de mapeado de textura diferentes que podrían entonces mezclarse ambas con un mezclador 602. Ya que el mapeado de textura 500 y la unidad de entorno de textura 600 en el modo de realización de ejemplo son ambas operaciones multitarea/multietapa, una secuencia de operaciones directa/indirecta puede proporcionarse basada en la salida del cálculo de iluminación 302 para proporcionar una variedad de efectos interesantes y complicados.

#### **Ejemplo de implementación de la tubería de iluminación**

**[0059]** La figura 11 muestra un ejemplo de diagrama de bloques de un cálculo de iluminación 302. En este ejemplo, el cálculo de iluminación 302 se elabora con la unidad de transformación 300 en

respuesta a la información recibida del procesador de órdenes 200. Esta información puede provenir de una variedad de fuentes incluyendo el procesador principal 110 y la memoria principal 112 (ver figura 4).

5 **[0060]** En el modo de realización de ejemplo, la unidad de transformación 300 incluye una sección de control maestro 320, una tubería de cálculo de iluminación 322, un acumulador 324 y, con el objetivo de generar textura, un conversor de enteros a puntos flotantes 308. El controlador maestro 320 recibe definiciones de iluminación y definiciones de vértice desde el procesador de órdenes 200 y, tras un almacenamiento/registro apropiado, proporciona información asociada a la tubería de cálculo de iluminación 322. La tubería de iluminación 322 elabora un cálculo de  
10 iluminación en la siguiente forma básica:

$$C_n = C_{material} [A_c + \sum_{i=0} L(c) \cdot A]$$

donde  $C_n$  define la salida de la tubería del cálculo de iluminación,  $C_{material}$  define el color del material,  $A_c$  define el color de ambiente,  $L(c)$  define el componente de iluminación difusa o especular, y  $A$  define la atenuación por distancia. Este cálculo computa un triplete RGB ya que la  
15 tubería de cálculo de iluminación 322 del modo de realización de ejemplo puede llevar a cabo dos de esos cálculos en paralelo. Debido a esta característica de cálculo paralelo, uno de los cálculos de iluminación puede utilizarse para efectos de iluminación acromática para un objeto y otro cálculo de iluminación puede utilizarse para efectos de iluminación cromática en el mismo objeto. La característica del cálculo paralelo permite efectos como una iluminación "libre" del dibujo  
20 animado en el sentido en que no se utiliza más tiempo de procesamiento en generar una superficie de polígono sombreada con efectos de iluminación de dibujo animado que en generar una superficie de polígono sombreada sin dichos efectos de iluminación de dibujo animado. En el modo de realización de ejemplo, el valor  $C_n$  generado por la tubería del cálculo de iluminación 322 puede ser positivo o negativo - con la iluminación negativa permitiendo la retroiluminación y otros  
25 efectos interesantes.

**[0061]** En el modo de realización de ejemplo, una salida  $C_n$  de la tubería de cálculo de iluminación 322 se acumula en un acumulador de enteros 324. La salida acumulada, en el modo de realización de ejemplo, se convierte en un punto flotante mediante el bloque de conversión 308 y se proporciona como coordenadas de textura a la unidad de textura 502 para operaciones de  
30 mapeado de textura. El modo de realización de ejemplo puede proporcionar un filtrado de textura como parte de la operación de mapeado de textura 306. El programador de aplicación debe ser cuidadoso en cuanto al modo elegido para el filtrado de textura al utilizar el cálculo de iluminación 302 para generar coordenadas de textura. Pueden surgir problemas debido al hecho de que los ejes  $s$  y  $t$  en el modo de realización de ejemplo se utilizan para factores independientes, pero  
35 ciertos tipos de filtrado de textura se aplican a ambos ejes de manera simultánea. Una solución es duplicar, dentro del mapa de textura, cada entrada a lo largo del eje  $t$  con tal de asegurarse de que la interpolación entre los valores  $t$  adyacentes no tiene ningún efecto sobre la salida final. Por lo tanto, un filtrado de textura unidimensional para valores adyacentes de  $t$  será idéntico. El filtrado

mipmap (multimapa) con múltiples niveles de detalles también puede tener lugar, pero este también puede llevarse a cabo para reducir el número de texturas unidimensionales dentro de un mapa de textura bidimensional dado.

5 **[0062]** En el modo de realización de ejemplo, otra complicación aparece por el método de convertir el valor de color del cálculo de iluminación 302 en coordenadas de textura. El valor de color producido por el cálculo de iluminación 302 en el ejemplo de implementación es un entero de 8-bits en el intervalo de 0-255. El bloque conversor 308 convierte este valor entero en un número de punto flotante dividiéndolo entre 255. Sin embargo, el valor se convierte en una coordenada de textura multiplicándolo por el tamaño de textura en el modo de realización de ejemplo. Este  
 10 proceso conlleva que el programador de aplicación debe tener especial cuidado con la coordenada de textura para elegir la textura unidimensional. Siempre que el tamaño de textura sea de 256 y el filtrado de textura sea GX\_NEAR, este se mapeará de la siguiente manera:

Valor de color	Coordenada de textura convertida	Valor más cercano
0	0	0
64	64,251	64
127	127,498	127
128	128,502	129
192	192,753	193
254	254,996	255
255	256	255

**[0063]** Debido al proceso de conversión, el valor de coordenada 128 se salta, y los valores de color 254 y 255 se mapean con el mismo valor de coordenada. Siempre que el tamaño de textura sea de 256 y el filtrado sea GX\_LINEAR, tendremos los siguientes mapeados:

Valor de color	Coordenada de textura convertida	Coordenadas consultadas
0	-0,5	0, 0
1	0,504	0, 1
2	1,508	1, 2
3	2,512	2, 3
4	3,516	3, 4
126	125,994	125, 126
127	126,998	126, 127
128	128,002	128, 129
253	253,492	253, 254
254	254,496	254, 255
255	255,5	255, 255

**[0064]** Resulta seguro utilizar  $n^2$  para convertir una tabla ID en un valor de color en el ejemplo de implementación. Sin embargo, las texturas unidimensionales deben almacenarse de manera no

directa en la textura 2D. La tabla para  $n=0$  debe almacenarse en la coordenada 0 (únicamente), las tablas para  $n=1$  hasta  $n=63$  deberían almacenarse en  $n*2-1$  y  $n*2$ , y las tablas para  $n=64$  hasta  $n=127$  deberían almacenarse en  $n*2$  y  $n*2+1$ . La coordenada 127 puede quedarse vacía ya que normalmente no se accederá a ella en esta implementación en particular. Otras implementaciones podrían evitar este problema, o podrían presentar otros problemas de conversión diferentes.

### **Descripción más detallada del ejemplo de la unidad de transformación 300**

**[0065]** Las figuras 12-20 muestran un ejemplo adicional, un modo de realización más detallado de la unidad de transformación 300 incluyendo una tubería de iluminación utilizada para el cálculo de iluminación 302. En el modo de realización de ejemplo, la unidad de transformación 300 admite la iluminación en hardware como un cálculo por vértice. Esto significa que un valor de color (RGB) puede calcularse para cada vértice iluminado, y que estos colores pueden interpolarse de manera lineal sobre la superficie de cada triángulo iluminado (conocido como sombreado Gouraud). La unidad de transformación 300 en este ejemplo de modo de realización es totalmente compatible con los focos de iluminación difusa locales, y parcialmente con la iluminación especular en el infinito.

**[0066]** La unidad de transformación 300 en el ejemplo del modo de realización es compatible con la atenuación difusa. Esto significa que la parte delantera del objeto puede ser más brillante que los laterales, y la parte posterior más oscura. La unidad de transformación 300 es compatible con las normales a los vértices para proporcionar atenuación difusa. Para cada vértice, la normal del vértice (N) se compara frente al vector entre el vértice y la posición de iluminación.

**[0067]** El modo de realización de ejemplo de la unidad de transformación 300 también soporta luces locales. Las luces locales tienen una posición dentro del mundo y posiblemente también una dirección. Cada luz se define por tener una posición. Utilizando la posición de cada vértice y la posición de luz, la unidad de transformación 300 puede llevar a cabo una distancia de atenuación por vértice. Esto significa que se puede disminuir el brillo de la luz que resplandece sobre un objeto a medida que el objeto se aleja de la luz.

**[0068]** La unidad de transformación 300 en el modo de realización de ejemplo también es compatible con iluminación direccional. Soporta desde intervalos de luces no direccionales hasta efectos direccionales sutiles, o hasta focos de luz altamente direccionales. Estos efectos son compatibles gracias a la atenuación angular. Por lo tanto, los vértices directamente "en el haz" de la luz pueden hacerse más brillantes que los vértices fuera del haz o detrás de la luz.

**[0069]** Las luces difusas locales pueden ser tanto de distancia atenuada como de ángulo atenuado. Al programar la ecuación de iluminación adecuada, es posible para un programador de aplicación obtener los valores de atenuación como una salida de color o alfa (o, en el caso de la generación de coordenada de textura, como una coordenada de textura).

**[0070]** En el ejemplo de implementación, la unidad de transformación soporta ocho luces físicas. El programador de aplicación puede describir los parámetros de atenuación, la posición, la dirección y el color para cada iluminación. El programador de la aplicación también puede

controlar hasta cuatro canales de color físicos que acumulan el resultado de la ecuación de iluminación. Al asociar luces con canales, el programador de la aplicación puede elegir sumar el efecto de múltiples luces por vértice, o combinarlas después en una unidad de entorno de textura 600. El número de canales disponibles para la unidad de entorno de textura 600 lo establece el programador de la aplicación. En algunos casos (p. ej., al utilizar un canal de color para generar coordenadas de textura), un canal de iluminación se calcula pero no se extrae como un color u opacidad. Como se ha comentado antes, la unidad de transformación 300 se canaliza para calcular dos canales de color simultáneamente, pero quizás proporcione solo un canal de color directamente a la unidad de entorno de textura 600 para mezclarse. El otro canal de color puede proporcionarse en forma de coordenadas de textura a la unidad de textura 500.

**[0071]** Cada canal de color permite la atenuación de la fuente de color en la selección. Una máscara de iluminación asocia hasta cinco luces con el canal. Vemos el canal de la figura 12 que muestra el ejemplo de asociación de hasta ocho luces diferentes con cualquiera de los dos canales de color y los dos canales alfa para proporcionar dos salidas independientes, una de las cuales puede convertirse en coordenadas de textura.

**[0072]** Como se muestra en la figura 12, la unidad de transformación 300 en el modo de realización de ejemplo genera dos colores RGBA (color 0 y color 1). Cada salida de color tiene dos funciones de iluminación: una para RGB y una para alfa, para un total de cuatro funciones de iluminación por polígono por vértice. Cuatro de dichas luces permiten una variedad de efectos de iluminación como, por ejemplo, la multitexturización. Cada función puede incluir un color de material en operaciones basadas en un color ambiental global y el estado de ocho luces conservadas en la unidad de transformación 300. Las ecuaciones permiten la atenuación difusa, especular y de foco de luz. La trayectoria de los datos de iluminación se diseña para ser óptima para focos de luz difusos locales en el ejemplo del modo de realización preferido, pero también es posible generar reflexiones especulares y/o factores de atenuación para una iluminación basada en textura con la misma trayectoria de datos. Como se ha descrito con anterioridad, los colores de iluminación también pueden utilizarse para generar coordenadas de textura.

**Ejemplo de cálculo de iluminación**

**[0073]** Las ecuaciones de iluminación de ejemplo llevadas a cabo por la unidad de transformación 300 son las siguientes:

$$C = \begin{cases} \text{Material}_c \begin{bmatrix} R \\ G \\ B \end{bmatrix} \times \text{Func.Ilum} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, & \text{si } C = \text{Color}_{0_{RGB}}, \text{Color}_{1_{RGB}} \\ \text{Material}_c [A] \times \text{Func.Ilum}_c [A], & \text{si } C = \text{Color}_{0_{Alfa}}, \text{Color}_{1_{Alfa}} \end{cases}$$

$$\text{Material}_c = \text{MaterialFuente}_c = \text{REGISTRO? ColorVértice}_c : \text{RegMaterial}_c$$

$$\text{Func.Ilum}_c = \begin{cases} 1.0, & \text{si Func.Ilum.Habilitada}_c = \text{FALSO} \\ \text{Ilum}_c, & \text{si Func.Ilum.Habilitada}_c = \text{VERDADERO} \end{cases}$$

$$Illum_c = \text{Ajustar} \left( Amb_c + \frac{\text{Entero con Signo}}{\text{Signo}} \left( \sum_{i=0}^{i=7} \text{Habilita}_c(i) \text{Aten}_c(i) \text{Aten.Difus}_c(i) \text{Color}_i \right) \right)$$

$$Amb_c = (\text{AmbFuente}_c = \text{REGISTRO? ColorVértice}_c : \text{RegAmbiente})$$

5 **[0074]** En el modo de realización de ejemplo, los colores del material y del ambiente global pueden extraerse de un registro o en la forma de un color de vértice desde el procesador de órdenes 200. Una aplicación que requiere más de ocho luces puede calcular la iluminación desde las luces adicionales, basándose en el software que se ejecuta en el procesador principal 110, para que el resultado sea uno de los colores de vértice, y establecer el registro de fuente ambiental para utilizar ese color de vértice. También es posible, si se desea, que el procesador principal 110 calcule cualquier o todos los cálculos de iluminación utilizando software y que proporcione esos cálculos resultantes extraídos en forma de colores de vértice a la unidad de transformación 300 para convertirlos en coordenadas de textura; o el procesador principal puede pasar las coordenadas de textura calculadas a la unidad de textura 500 con el propósito de mapear la textura.

15 **[0075]** Al deshabilitar el parámetro "Func.Illum", el color del material pasa a través de la unidad de transformación 300 sin modificarse. Esto puede utilizarse para permitir al procesador principal 110 seleccionar directamente, basándose en el programa de aplicación de control de software, qué textura utilizar de una pluralidad de texturas unidimensionales en una operación de mapeado de textura que reacciona a una salida de función de iluminación acromática.

20 **[0076]** Cualquiera o la totalidad de las ocho luces disponibles pueden habilitarse en cada función de iluminación. La suma de la iluminación por luz se fija a [-1,1] en el modo de realización de ejemplo, y se convierte en complemento de 2 enteros antes de añadir el término de ambiente global. Debido a que la iluminación total se fija a [0, 1], un material no puede volverse más brillante a través de la iluminación en el modo de realización de ejemplo. Si se desea este efecto, 25 los colores de iluminación pueden pre-multiplicarse por el color del material, y el color del material en la ecuación puede establecerse a 1,0. Otras configuraciones pueden proporcionarse en otros ejemplos de implementación para permitir que los materiales se vuelvan más brillantes a través de la iluminación.

30 **[0077]** En el modo de realización de ejemplo, las ecuaciones alfa pueden utilizarse cuando los resultados monocromos son suficiente. Los valores alfa resultantes pueden combinarse con otros valores RGB y alfa en una unidad de entorno de textura 600 en el modo de realización de ejemplo. Por ejemplo, es posible poner el resultado difuso en la salida de color para el canal de color 0, un resultado especular en la salida alfa del canal de color 0, y hacer que la unidad de entorno de textura 600 multiplique el color difuso por el color de textura y añada un reflejo 35 especular (por ejemplo, blanco) en una única etapa de la unidad de entorno de textura 600.

**[0078]** La flexibilidad se proporciona en la función de atenuación difusa en el modo de realización de ejemplo, más allá de la orden  $(N H)$  ajustar\_a  $[0,1]$ . La atenuación puede evitarse para las

ecuaciones de iluminación que no tienen una propiedad difusa o se dejan sin ajustar. Los productos escalares sin ajustar permiten generar una textura basada en la iluminación para controlar la iluminación a 180°. Debido a que la iluminación se ajustará en el ejemplo de modo de realización, las funciones de iluminación con signo pueden escalarse y polarizarse de manera apropiada utilizando colorc y ambc. El modo de realización de ejemplo proporciona tres ejemplos de funciones de atenuación difusa que una ecuación de iluminación puede utilizar:

$$Aten.Difusa_c(i) = \begin{cases} 1.0, & \text{si Selec.Aten.Difus}_c = \text{NINGUNA} \\ \hat{N} \cdot \hat{L}_i, & \text{si Selec.Aten.Difus}_c = \text{con SIGNO (sólo si FOCO DE LUZ)} \\ Ajuste0(\hat{N} \cdot \hat{L}_i), & \text{si Selec.Aten.Difus}_c = \text{ajuste (sólo si FOCO DE LUZ)} \end{cases}$$

**[0079]** El ángulo de atenuación lógico en el ejemplo propuesto se calcula a través de un polinomio de segundo orden basado en el producto escalar del vector iluminación sobre el vértice y el vector de dirección de iluminación. Se consigue una fuerte caída al extrapolar los términos cuadrado y lineal. El ajuste se utiliza para evitar los valores negativos producidos por la extrapolación de los ángulos fuera de un ángulo de foco. El rango de atenuación puede llevarse a cabo mediante la inversa de otro polinomio de segundo orden. El valor de distancia, d, es la longitud del vector desde el vértice hasta la posición de iluminación. Una ecuación de iluminación con sólo distancia de atenuación puede utilizarse con iluminación basada en textura para simular la caída de iluminación basada en la distancia de texturas proyectadas u otras texturas.

**[0080]** Esta misma lógica puede utilizarse para aproximar la atenuación especular común, (N · H)s para una fuente de iluminación paralela. Los vectores de iluminación sobre el vértice y los vectores de dirección de luz pueden reemplazarse con los vectores normales y de medio ángulo respectivamente. La atenuación puede evitarse en una función de iluminación para permitir que las fuentes de punto de iluminación no atenuadas (p. ej., omnidireccional) se utilicen tanto en una ecuación difusa como especular. En este ejemplo, los coeficientes de atenuación del ángulo de iluminación pueden utilizarse para la ecuación especular, y la atenuación se desactiva en la ecuación difusa. Las ecuaciones especular y difusa en el ejemplo de modo de realización se combinan en el entorno de textura 600 al contrario que en el interior de la unidad de transformación 300 en el ejemplo propuesto, pero también son posibles otras implementaciones.

**[0081]** Los siguientes son detalles adicionales del ejemplo de función de atenuación:

$$Aten_c = AtenHabilitada_c \cdot \frac{Ajuste0(a_2 \cdot AAten_c(i)^2 + A_{1_i} \cdot AAten_c(i) + A_{0_i})}{K_2 \cdot d_c^2(i) + k_1 \cdot d_c(i) + k_0} : 1.0$$

$$AAten_c(i) = \begin{cases} \hat{N} \cdot \hat{L} > 0 \text{ Ajustar}_0(\hat{N} \cdot H_i) : 0, \text{ if } AtenSelec_c = \text{ESPECULAR} \\ Ajuste0(\hat{L} \cdot L_{dir}), \text{ if } AtenSelec_c = \text{FOCO DE LUZ} \end{cases}$$

$$d_c(i) = \begin{cases} \hat{N} \cdot \hat{L} > 0 ? Ajuste0(\hat{N} \cdot H_i) : 0, \text{ if } AtenSelec_c = \text{ESPECULAR} \\ \sqrt{\hat{L}_i \cdot L_i}, \text{ if } AtenSelec_c = \text{FOCO DE LUZ} \end{cases}$$

**Diagrama de bloques del ejemplo de implementación de la unidad de transformación 300**

**[0082]** La figura 13 es un diagrama de bloques de un ejemplo de implementación de la unidad de

transformación 300 del sistema 50. La unidad de transformación 300 en este ejemplo de modo de realización incluye tres secciones principales:

- sección superior de la tubería (TOP) 330,
- sección de iluminación 332,
- 5 • sección inferior de la tubería (BOP) 334.

**[0083]** La sección superior de la tubería 330 en el modo de realización de ejemplo incluye una sección de iluminación 332, un registro de contexto de matriz 336, una entrada de memoria *FIFO* 338, una unidad de producto escalar 340, un bloque de proyección 342, un bloque de punto de textura dos 346, un detector de recorte 348, y un bloque de desacoplo de *FIFO* 350. En el modo  
10 de realización de ejemplo, la sección superior de la tubería (TOP) 330 elabora las siguientes funciones de ejemplo:

- transformación del vértice (3 productos escalares),
- transformación de la Normal (3 productos escalares),
- transformación de la textura (2 or 3 productos escalares por textura),
- 15 • transformación de proyección (3 productos escalares simplificados); y  
una sección de iluminación 332 elabora los siguientes ejemplos de funciones de canal de color:
- cálculo difuso del canal de color 0 (1 producto escalar N·L),
- cálculo difuso del canal de color 0 (1 producto escalar L2 por iluminación),
- 20 • cálculo difuso del canal de color 0 (1 producto escalar N2).

**[0084]** Además, la entrada *FIFO* recibe descriptores de vértice por vértice especificando, por ejemplo, la siguiente información:

- información de geometría XYZ,
- información de vector Normal  $N_x$ ,  $N_y$ ,  $N_z$ ,
- 25 • color 0 RGBA por vértice,
- color 1 RGBA por vértice,
- vector binormal  $T_x$ ,  $T_y$ ,  $T_z$ ,
- vector binormal  $B_x$ ,  $B_y$ ,  $B_z$
- datos  $T_0$  de textura 0,
- 30 • datos  $S_n$ ,  $T_n$  de textura 1 a n,

**[0085]** Se proporciona información por vértice apropiada a la sección de iluminación 332 para permitir los cálculos de iluminación.

**[0086]** La sección de iluminación 332 en el modo de realización de ejemplo mostrado en la figura 13 incluye una memoria de parámetros de iluminación 352, una memoria de la Normal 354 y una  
35 trayectoria de datos de iluminación 356.

**[0087]** La memoria del parámetro de iluminación 352 almacena varios parámetros de iluminación utilizados por la trayectoria de datos de iluminación 356. La memoria del parámetro de iluminación 352 contiene por lo tanto toda la información de iluminación (por ejemplo, vectores de iluminación, parámetro de iluminación, etc.). Tanto el estado global 30 como el estado ambiental se almacenan

en esta memoria en el modo de realización de ejemplo. En el modo de realización de ejemplo, cada palabra se escribe en 32-bits, pero sólo las veinte más importantes se guardan. Cada localización ocupa tres palabras, con un mínimo de tamaño de escritura de palabra de tres palabras en el modo de realización de ejemplo. A continuación se muestran ejemplos de contenidos de una memoria de parámetro de iluminación 352:

5

Dirección de registro	Definición	Configuración
0/0600	Reservado	
0x0601	Reservado	
0x0602	Reservado	
10 0x0603	Iluminación0	32b. RGBA (8b/comp)
0x0604	Iluminación0A0	20b: cos. aten. $A_0$
0x0605	Iluminación0A1	20b: cos. aten. $A_1$
0x0606	Iluminación0A2	20b: cos. aten. $A_2$
15 0x0607	Iluminación0K0	20b: dist. aten. $K_0$
0x0608	Iluminación0K1	20b: dist. aten. $K_1$
0x0609	Iluminación0K2	20b: dist. aten. $K_2$
0x060a	Iluminación0Lpx	20b: pos. luz x, o inf ldir x
0x060b	Iluminación0Lpy	20b: pos. luz x, o inf ldir x
0x060c	Luz0Lpz	20b: pos. luz x, o inf ldir x
20 0x060d	Luz0Dx/Hx	20b: dir. luz x, o 1/2 ángulo x
0x060e	Luz0Dy/Hy	20b: dir luz x, o 1/2 ángulo x
0x060f	Luz0Dz/Hz	20b: dir luz z, o 1/2 ángulo x
0x0610-0x067f	Datos (n) de Luz	Ver datos de Luz0
0x0680-0x07ff	No usado	Reservado

25

**[0088]** Una visión general de la sección de iluminación 332 del modo de realización de ejemplo se muestra en la figura 14. Brevemente, la sección de iluminación 332 elabora los siguientes cálculos de iluminación local:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \text{Material}_0 \begin{bmatrix} Mr \\ Mg \\ Mb \end{bmatrix}$$

$$\left( \text{Ambient}_0 \begin{bmatrix} Ra \\ Ga \\ Ba \end{bmatrix} + \frac{\text{FuncCosAt}_0}{\text{FuncDisAt}_0} \text{Ajustar} ( \text{Normal} \cdot \text{Illum}_i ) \times \text{Difuso}_i \begin{bmatrix} Rd \\ Gd \\ Bd \\ 0.0 \end{bmatrix} \right)$$

30

Esto requiere un cálculo de vector de iluminación local por vértice:

$\text{Illum}_i$       Normalizada (PosIllum<sub>i</sub>)      Vértice Ojo Espacio

$$Light_i = Normalized(LightPos_i \begin{bmatrix} Lx \\ Ly \\ Lz \end{bmatrix} - EyeSpaceVertex \begin{bmatrix} Xh \\ Yh \\ Zh \end{bmatrix})$$

tras el cual, se necesita calcular:

$$AtenDifusa_0 = \text{Condicional\_ajustar\_a\_0} (Illum_i \cdot \text{Normal Ojo Espacio})$$

**[0089]** Por lo tanto, el cálculo de la atenuación difusa requiere el vector de iluminación y el vector Normal ojo-espacio. Ajustar a 0 puede ser condicional en una señal interna (p.ej. puede limitarse a 1,0 o no limitarse en absoluto) en el ejemplo propuesto. Es posible permitir que las luces de difusión tengan productos escalares negativos. Una unidad de producto escalar de iluminación puede calcular el vector Normal ojo-espacio a la vez que la unidad de transformación de producto escalar calcula la conversión de geometría ojo-espacio. Entonces, una unidad de producto escalar de iluminación puede calcular los productos escalares N·L, L2 y N2. Después de esto, el ejemplo propuesto normaliza los resultados para proporcionar información normalizada de un vector Normal de iluminación. Tras la normalización, los resultados intermedios de la normalización (p.ej. la distancia cuadrática y la distancia y el coseno N·L) se utilizan para calcular las ecuaciones de atenuación así como los factores de normalización. Entonces, el vector de iluminación ambiental se multiplica. El triplete resultante es el componente de atenuación difusa por iluminación. Este se convierte a 2 enteros complementarios, y los tres valores resultantes se acumulan en un acumulador de enteros. Este acumulador añade los términos ambientales y cualquier otro término de difusión desde otras luces. La suma final es entonces el color por vértice (en el ejemplo propuesto, un formato RGB de 8bits ajustado de 0 a 255).

**[0090]** La figura 15 muestra un diagrama de ejemplo del bloque de punto de vector/ vector Madd 357 mostrado en la figura 14. Si la misma unidad de producto escalar se utiliza para calcular la transformación N y todos los demás productos escalares, entonces las representaciones pueden rebajarse de alguna manera y la programación de la iluminación se vuelve difícil ya que cada iluminación necesita reutilizar el mismo elemento de trayectoria de datos múltiples veces. Esto podría provocar problemas de control complejos. Con tal de simplificar el diseño, podemos implementar una segunda unidad de producto escalar ajustada para separar los diferentes cálculos. La primera unidad de producto escalar puede calcular la transformación Normal a la vez que la otra unidad de producto escalar calcula la transformación de vértice. Una suma de vector puede entonces calcular el vector de iluminación para la iluminación local. El segundo producto escalar puede calcular N2, L2, N·L y N·H (para luces especulares). Esto se puede conseguir en el modo de realización de ejemplo de forma totalmente vectorial sin retroalimentación. Ver la figura 15.

**[0091]** La figura 16 muestra un ejemplo de implementación del normalizador 360 mostrado en la figura 14. Una vez que los valores del vector de iluminación se calculan, el modo de realización de ejemplo normaliza los resultados y calcula la atenuación. El normalizador del modo de realización de ejemplo 360 acepta las siguientes entradas:

- Coseno de atenuación: L·Dir, 1/raíz cuadrada (L)

- Distancia de atenuación:  $L_2, L$
- Factor difuso:  $N \cdot L, N_2, L_2,$

y calcula la distancia de atenuación utilizando el siguiente paso ejecutado por el ejemplo normalizado 360 mostrado en la figura 16:

- 5           1. Calcular  $K_2d_2$
2. Calcular  $K_2d_2 + K_0$
3. Calcular  $d$
4. Calcular  $K_1d$
5. Calcular  $D_2D_2 + K_1d + K_0$
- 10          6. Calcular  $1/(K_2d_2 + K_1d + K_0)$

**[0092]** La figura 17 muestra un ejemplo de unidad de Distancia de atenuación 362. El ejemplo de implementación de la Distancia de atenuación 362 calcula el Coseno de atenuación utilizando los siguientes pasos de ejemplo:

1. Calcular  $\text{Cos} = \text{Ajuste}_0 (L \cdot L_{dir})$
- 15          2. Calcular  $\text{Cos}^2$
3. Calcular  $A_2\text{Cos}^2$
4. Calcular  $A_1\text{Cos}$
5. Calcular  $A_1\text{cos} + A_0$
6. Calcular  $A_2\text{Cos} + A_1\text{Cos} + A_0$
- 20          7. Calcular  $\text{Ajuste}_0(A_2 + \text{Cos}^2 + A_1\text{Cos} + A_0)$

**[0093]** La figura 18 muestra un ejemplo de implementación de la escala de iluminación 364 mostrado a continuación:

- 1/raíz cuadrada ( $N_2$ )
- 1/raíz cuadrada ( $L_2$ )
- 25 •            $N \cdot L$  (raíz cuadrada ( $N_2$ ) x raíz cuadrada ( $L_2$ )).

**[0094]** Algunas unidades (p.ej. 1/raíz cuadrada y algunos otros multiplicadores) pueden compartirse entre las diferentes implementaciones de trayectoria de datos mostradas en las figuras de la 15 a la 18. Por lo tanto, una implementación podría necesitar aproximadamente de la orden de dieciocho multiplicadores y diez sumadores. La inversión y 1/raíz cuadrada pueden ejecutarse utilizando una tabla de búsqueda o un simple pase por el interpolador Newton-Raphson o algún otro interpolado. Los cálculos de escritura alfa y especular no requieren ningún hardware adicional en el modo de realización del ejemplo ya que simplemente son cambios en la ecuación de iluminación.

**[0095]** La figura 19 muestra un ejemplo de implementación de acumulador de enteros 336. Cuando un nuevo atributo llega para cada color, ese color se acumula. Una vez que todas las luces se acumulan para un color, y se añade el ambiente (p.ej. desde el color de vértice *FIFO* o desde un registro), entonces el color del material se multiplica. El color RGB/A final se acumula entonces en un registro de acumulación/formato mediante la construcción del color final y la generación de textura. Una vez que se calcula el color/textura, se escribe en una salida *FIFO*. El

control puede seguir estas líneas generales:

Si (nuevo atributo == nueva iluminación para color X)

Acumular nueva iluminación en el color X incrementar el número de luces para el color X

Fin

5

Si (número de luces para el color X es máximo)

Transferir el color al multiplicador

Ejecutar multiplicación de material

Fin

10

Si (el color X está en estado de multiplicar)

Primer color acumulado X

Fin

15

Si (la acumulación de color X está completa ((RGBX o AXX o RGBA))

El formato de color X para color, si el color está habilitado

El formato de color X para textura, si la textura está habilitada y el color deshabilitado o el color ya se ha cargado

La carga de salida *fifo* con los datos formateados incrementa el cálculo de color/textura

20

Fin

Si (el cálculo de color y el cálculo de textura son máximos)

Escribir datos en Sección inferior de la tubería Pop fifo

Fin

25

**[0096]** La figura 20 muestra un ejemplo de diagrama de etapas de transición utilizadas para controlar la tubería de iluminación 332 en el modo de realización de ejemplo. El árbitro lógico decide la secuencia de los problemas que se envían a la trayectoria de datos de iluminación. El árbitro del modo de realización de ejemplo sólo puede enviar un nuevo problema cada cuatro u

30 ocho ciclos, dependiendo del problema. La lista de problemas posibles incluye:

- el cálculo de iluminación para un color/alfa específico
- cálculo de mapa de rugosidad.

**[0097]** Para los problemas mencionados, el árbitro deja caer un atributo en la tubería de control cada 4/8 ciclos. Cada uno de estos atributos sigue las instrucciones de ejecución para el problema

35 en la tubería de iluminación 332. Al final de la tubería, los atributos se utilizan para incrementar las cuentas locales que se utilizan para determinar si los colores/texturas están completos/as.

### **Otros ejemplos de implementaciones compatibles**

**[0098]** Algunos de los componentes de sistema 50 descritos arriba podrían implementarse como

40 otra configuración de videoconsola doméstica descrita arriba. Por ejemplo, se podría utilizar una aplicación de gráficos u otro software escrito para un sistema 50 sobre una plataforma con una configuración diferente que emula el sistema 50 o que es de otra forma compatible con este. Si la otra plataforma puede emular, simular y/o suministrar con éxito alguno o todos los recursos de hardware y software del sistema 50, entonces la otra plataforma será capaz de ejecutar el

software con éxito.

**[0099]** Como un ejemplo, un emulador puede proporcionar una configuración (plataforma) de hardware y/o software que es diferente a la configuración (plataforma) de hardware y/o software del sistema 50. El emulador de sistema puede incluir componentes de software y/o hardware que emulan o simulan alguno o todos los componentes de hardware y/o software del sistema para el cual se escribió el software de la aplicación. Por ejemplo, el emulador de sistema puede comprender un ordenador digital de uso general como un ordenador personal, que ejecuta un programa emulador de software que simula el hardware y/o firmware del sistema 50.

**[0100]** Algunos ordenadores digitales con uso general (p.ej. los ordenadores personales IBM o Macintosh y compatibles) están ahora equipados con tarjetas gráficas 3D que proporcionan tuberías de gráficos 3D cumpliendo con DirectX u otras órdenes de APIs de gráficos 3D estándar. También pueden equiparse con tarjetas de sonido estereofónico que proporciona sonido estereofónico de alta calidad basado en un conjunto de órdenes de sonido estándar. Dichos ordenadores personales multimedia equipados con hardware que ejecutan un emulador de software pueden tener la suficiente calidad como para aproximarse a la calidad de gráficos y sonido del sistema 50. El emulador de software controla los recursos de hardware en la plataforma del ordenador personal para simular el procesamiento, los gráficos 3D, el sonido, la capacidad periférica y otras de una plataforma de consola de videojuegos doméstica para la cual el programador de juego escribió el software del juego.

**[0101]** La figura 21A ilustra un ejemplo del proceso de emulación global utilizando una plataforma anfitrión 1201, un componente emulador 1303, y una imagen binaria ejecutable con software de juego proporcionada en un medio de almacenamiento 62. El anfitrión 1201 puede ser un dispositivo de computación digital con uso general o específico como, por ejemplo, un ordenador personal, una consola de videojuegos, o cualquier otra plataforma con suficiente potencia de computación. El emulador 1303 puede ser software y/o hardware que se ejecuta sobre la plataforma anfitrión 1201, y proporciona a tiempo real una conversión de órdenes, datos y otra información del medio de almacenamiento 62 de forma que pueda procesarse con el anfitrión 1201. Por ejemplo, el emulador 1303 extrae instrucciones "fuente" de las instrucciones de programa de la imagen binaria con la intención de ejecutarlas mediante el sistema 50 desde el medio de almacenamiento 62 y convierte estas instrucciones de programa en un formato objetivo que puede ejecutarse o procesarse de otra manera mediante el anfitrión 1201.

**[0102]** Como un ejemplo, en el caso donde el software se escribe para su ejecución sobre una plataforma utilizando un PowerPC IBM u otro procesador específico y el anfitrión 1201 es un ordenador personal utilizando un procesador diferente (p.ej. Intel), el emulador 1303 extrae una instrucción o una secuencia de instrucciones del programa de imagen binaria desde el medio de almacenamiento 62 y convierte estas instrucciones de programa en una o más instrucciones del programa de imagen binaria equivalente a Intel. El emulador 1303 también extrae y/o genera órdenes de gráficos y órdenes de audio para procesar mediante el procesador de gráficos y de audio 114, y convierte estas órdenes en un formato o formatos que pueden procesarse mediante gráficos de hardware y/o software y recursos de procesamiento de audio disponibles en el

anfitrión 1201. Como ejemplo, el emulador 1303 puede convertir estas órdenes en órdenes que pueden procesarse mediante el hardware del anfitrión 1201 específico de gráficos y/o de sonido (p.ej. utilizando un estándar DirectX, OpenGL y/o APIs de sonidos).

**[0103]** Un emulador 1303 utilizado para proporcionar alguna o todas las características del sistema de videojuegos descrito arriba también puede proporcionarse con una interfaz gráfica de usuario (GUI) que simplifica o automatiza la selección de varias opciones y modos de pantalla para ejecutar juegos utilizando el emulador. En un ejemplo, tal emulador 1303 también puede incluir funcionalidad mejorada en comparación con la plataforma anfitrión para la que el software se concibió originalmente.

**[0104]** La figura 21B ilustra un sistema anfitrión de emulación 1201 adecuado para su uso con emulador 1303. El sistema 1201 incluye una unidad de procesamiento 1203 y una memoria de sistema 1205. Un bus de sistema 1207 acopla varios componentes de sistema incluyendo una memoria de sistema 1205 en la unidad de procesamiento 1203. El bus de sistema 1207 puede ser cualquiera de los varios tipos de estructuras de bus incluyendo un bus de memoria o un controlador de memoria, un bus periférico, y un bus local utilizando cualquiera de la variedad de arquitecturas de bus. La memoria de sistema 1207 incluye memoria de solo lectura (ROM) 1252 y memoria de acceso aleatorio (RAM) 1254. Un sistema básico de entrada/salida (BIOS) 1256, conteniendo las rutinas básicas que ayudan a transferir la información entre elementos dentro del sistema del ordenador personal 1201, como durante el arranque, se almacena en la memoria ROM 1252. El sistema 1201 también incluye varias unidades de disco y medios legibles por ordenador asociados. Una unidad de disco duro 1209 lee desde y escribe sobre (normalmente fijado) un disco duro magnético 1211. Una unidad de disco magnético adicional 1213 (posible opcional) lee desde y escribe sobre un disco flexible u otro disco magnético 1215. Una unidad de disco óptico 1217 lee desde y, en algunas configuraciones, escribe sobre un disco óptico extraíble 1219, como un CD-ROM u otro medio óptico. La unidad de disco duro 1209 y la unidad de disco óptico 1217 están conectadas al bus de sistema 1207 mediante una interfaz de unidad de disco duro 1221 y una interfaz de unidad de disco óptico 1225 respectivamente. Las unidades de disco y sus medios asociados legibles por ordenador proporcionan un almacenaje no volátil de las instrucciones legibles por ordenador, las estructuras de datos, los módulos de programa, los programas de juego y otros datos para un sistema de ordenador personal 1201. En otras configuraciones, otros tipos de medios legibles por ordenador que pueden almacenar datos que sean accesibles por ordenador (p.ej. cassetes, tarjetas de memoria flash, discos de vídeo digital, cartuchos de Bernoulli, memorias de acceso aleatorio (RAMs), memorias de solo lectura (ROMs) y similares) también pueden utilizarse.

**[0105]** Un número de módulos de programa incluyendo un emulador 1303 pueden almacenarse en el disco duro 1211, el disco magnético extraíble 1215, el disco óptico 1219 y/o la memoria ROM 1252, y/o la memoria RAM 1254 de la memoria del sistema 1205. Dichos módulos de programa pueden incluir un sistema operativo que proporciona APIs de gráficos y de sonido, uno o más programas de aplicación, otros módulos de programa, datos de programa y datos de juego. Un usuario puede introducir órdenes e información en un sistema de ordenador personal 1201 a

través de dispositivos de entrada como un teclado 1227, un dispositivo puntero 1229, micrófonos, joysticks, controladores de juego, antenas parabólicas, escáneres, o similares. Estos y otros dispositivos de entrada pueden conectarse a la unidad de procesamiento 1203 a través de la interfaz del puerto serial 1231 que se asocia al bus de sistema 1207, pero puede conectarse mediante otras interfaces, como un puerto paralelo, un puerto de juego, un bus FireWire o un bus universal en serie (USB). Un monitor 1233 u otro tipo de dispositivo de visualización también está conectado al bus de sistema 1207 a través de una interfaz, como un adaptador de vídeo 1235.

**[0106]** El sistema 1201 también puede incluir un módem 1154 u otro medio de interfaz de red para establecer comunicaciones sobre una red 1152 como la de internet. El módem 1154, que puede ser interno o externo, está conectado al bus del sistema 123 a través de la interfaz de puerto en serie 1231. Una interfaz de red 1156 también puede incluirse para permitir que el sistema 1201 comunique con un dispositivo de cálculo remoto 1150 (p.ej. otro sistema 1201) a través de una red de área local 1158 (o dicha comunicación puede establecerse a través de una red de área amplia 1152 u otras trayectorias de comunicación como un medio de acceso telefónico y otras comunicaciones). El sistema 1201 incluirá típicamente otros dispositivos de salida periféricos, como impresoras y otros dispositivos periféricos estándar.

**[0107]** En un ejemplo, el adaptador de vídeo 1235 puede incluir un conjunto de chips de tubería de gráficos 3D que proporcionan una representación rápida de gráficos 3D en respuesta a las órdenes de gráficos 3D enviadas basadas en una interfaz del programador de aplicación de gráficos 3D estándar como Microsoft's DirectX 7.0 u otra versión. Un conjunto de altavoces estéreo 1237 también se conecta a un bus del sistema 1207 a través de una interfaz de generación de sonido como una "tarjeta de sonido" convencional que proporciona soporte hardware y software incorporado para generar un sonido estereofónico de alta calidad basado en órdenes de sonido proporcionadas por el bus 1207. Estas capacidades hardware permiten al sistema 1201 proporcionar suficiente calidad de gráficos y velocidad de sonido como para ejecutar el software almacenado en el medio de almacenamiento 62.

**[0108]** Mientras que la invención ha sido descrita en conexión con lo que en el presente se considera el modo de realización más práctico y preferido, debe entenderse que la invención no debe limitarse al modo de realización publicado, sino al contrario, pretende cubrir varias modificaciones y disposiciones equivalentes incluidas en el ámbito de las reivindicaciones adjuntas. Por ejemplo, como se apreciará, pese a que aquí se publica una iluminación de dibujo animado en particular, la invención no está limitada a la iluminación del dibujo animado, sino que, abarca una variedad de aplicaciones fotorealistas y no fotorealistas, basadas en textura y no basadas en textura y efectos que pueden conseguirse utilizando un parámetro de iluminación acromático.

**Reivindicaciones**

1. Un método de generación de imágenes gráficas en un sistema de generación de gráficos (50) del tipo que define al menos una función de iluminación por vértice para el sombreado de polígono, el método comprendiendo:  
5 el uso de la función de iluminación (302) para calcular al menos un parámetro acromático a utilizar en una modificación posterior (304) de un color u opacidad; y la representación de una imagen basada, al menos en parte, en el parámetro calculado; en el que dicho parámetro calculado tiene un valor negativo para la retroiluminación de un polígono.  
10
2. El método de acuerdo con la reivindicación 1, **caracterizado además porque** el parámetro comprende al menos una coordenada de textura (S).
3. El método de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado además porque** la función de iluminación (302) calcula dicho al menos un parámetro basado, al menos en parte, en la distancia de atenuación entre un vértice y una fuente de iluminación.  
15
4. El método de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado además porque** dicha función de iluminación (302) calcula dicho al menos un parámetro basado, al menos en parte, en el ángulo entre la fuente de luz en dirección al vértice y la normal de la superficie.
- 20 5. El método de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado además porque** dicha función de iluminación (302) calcula dicho al menos un parámetro basado, al menos en parte, en una posición de vértice del vector.
6. El método de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado además porque** dicha función de iluminación (302) define una función basada en la posición de superficie, la orientación de superficie y la posición de al menos una luz.  
25
7. El método de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado porque** dicha función de iluminación (302) puede utilizar una iluminación difusa o especular.
8. El método de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado porque** la etapa de representación comprende generar una textura para una superficie basada, al menos en parte, en dicho parámetro.  
30
9. El método de acuerdo con cualquiera de las reivindicaciones precedentes, **caracterizado además porque** dicha etapa de representación comprende el mapeado de textura utilizando una textura de una dimensión indexada por el mencionado al menos un **parámetro**.
10. El método de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado**

**porque** el método también incluye:

el uso de una función de iluminación (302) para generar al menos un valor de color u opacidad;  
 la conversión de dicho valor de color u opacidad en al menos una coordenada de textura (S, T);  
 el uso de dicha coordenada de textura en al menos una operación de mapeado de textura (306); y  
 5 el uso de los resultados de dicha operación de mapeado de textura (306) para modificar el color o  
 la opacidad de al menos una superficie visible en una imagen animada generada de manera  
 dinámica.

11. El método de acuerdo con la reivindicación 10, **caracterizado además porque** dicha  
 operación de mapeado de textura (306) comprende el mapeado de una textura unidimensional en  
 10 respuesta a dicha coordenada de textura.

12. El método de acuerdo con la reivindicación 10 u 11, **caracterizado porque** dicha operación  
 de mapeado de textura (306) comprende el mapeado de pinceladas en salidas de cálculo de la  
 iluminación dinámica.

13. El método de cualquiera de las reivindicaciones anteriores de la 10 a la 12, **caracterizado**  
 15 **además porque** dicha coordenada de textura (S,T) puede estar generada negativamente para  
 definir la retroiluminación de una imagen animada.

14. El método de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado**  
**porque** dicha función de iluminación (302) define una primera coordenada de textura (S, T) para  
 dicho mapeado de textura y una segunda coordenada de textura se deriva de una fuente diferente  
 20 a dicha función de iluminación (302).

15. El método de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado**  
**porque** dicha función de iluminación (302) comprende una función de iluminación difusa.

16. El método de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado**  
**porque** dicha función de iluminación (302) comprende una función de iluminación especular.

17. El método de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado**  
 25 **además porque** dicha función de iluminación (302) comprende una función de iluminación de  
 distancia de atenuación.

18. El método de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado**  
**porque** dicha función de iluminación (302) comprende una función de iluminación de foco de luz.

19. El método de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado**  
 30 **además porque** los parámetros generados pueden estar calculados en una plataforma anfitrión o  
 calculados utilizando la función de iluminación (302), como se requiera.

20. El método de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado**  
**además porque** un parámetro generado se utiliza para seleccionar un texel dentro de una textura

unidimensional (1D) y un segundo parámetro se utiliza para seleccionar qué textura 1D utilizar.

**21.** Un sistema de generación de gráficos del tipo que define al menos una función de iluminación por vértice (302) para el sombreado de polígono, el sistema comprendiendo:

un medio para utilizar la función de iluminación (302) para calcular al menos un parámetro acromático a utilizar en una modificación posterior (304) de un color u opacidad; y

un medio para la representación de una imagen basada, al menos en parte, en el parámetro calculado;

dicho parámetro calculado teniendo un valor negativo para la retroiluminación de un polígono.

**22.** El sistema de acuerdo con la reivindicación 21, comprendiendo además:

una tubería de datos de iluminación (300) que comprende al menos un primer y un segundo canal, dicha tubería de datos de iluminación (300) recibiendo la información por vértice y las definiciones de iluminación, y calculando parámetros de vértice iluminado en respuesta a dichas definiciones por vértice y de iluminación;

una unidad de textura (500) asociada a dicha tubería de datos de iluminación (300), dicha unidad de textura (500) realizando al menos una operación de mapeado de textura utilizando al menos una de dichas salidas de tubería de datos de iluminación (300) como al menos una coordenada de textura; y

un mezclador (700) asociado a dicha tubería de datos de iluminación (300) y a dicha unidad de textura (500), dicho mezclador (700) mezclando información cromática y/o de opacidad obtenida a partir de dicha tubería de datos de iluminación (300) con texels obtenidos a partir de dicha unidad de textura (500) para obtener una representación de imagen.

**23.** El sistema de acuerdo con la reivindicación 22, **caracterizado además porque** las salidas de la tubería de datos de iluminación (300) comprenden al menos una coordenada de textura.

**24.** El sistema de acuerdo con las reivindicaciones 22 o 23, **caracterizado además porque** la salida de la tubería de datos de iluminación (300) se calcula basada, al menos en parte, en la distancia de atenuación entre un vértice y una fuente de iluminación.

**25.** El sistema de acuerdo con cualquiera de las reivindicaciones de la 22 a la 24, **caracterizado además porque** dicha salida de la tubería de datos de iluminación (300) se calcula basada, al menos en parte, en el ángulo entre la fuente de luz en dirección al vértice y la normal de la superficie.

**26.** El sistema de acuerdo con cualquiera de las reivindicaciones de la 22 a la 25, **caracterizado además porque** dicha salida de la tubería de datos de iluminación (300) se calcula basada, al menos en parte, en una posición de vértice del vector.

**27.** El sistema de acuerdo con cualquiera de las reivindicaciones de la 22 a la 26, **caracterizado además porque** dicha salida de tubería de datos de iluminación (300) define una función basada

en la posición de superficie, la orientación de superficie y la posición de al menos una luz.

**28.** El sistema de acuerdo con cualquiera de las reivindicaciones anteriores de la 22 a la 27, **caracterizado además porque** dicha salida de la tubería de datos de iluminación (300) se calcula en forma de un parámetro negativo para la retroiluminación.

5 **29.** El sistema de acuerdo con cualquiera de las reivindicaciones anteriores de la 22 a la 28, **caracterizado además porque** dichas salidas de la tubería de datos de iluminación (300) pueden ser para iluminación difusa o especular.

**30.** El sistema de acuerdo con cualquiera de las reivindicaciones anteriores de la 22 a la 29, **caracterizado además porque** dicha unidad de textura (500) utiliza una textura unidimensional indexada por dicho al menos un parámetro.

**31.** El sistema de acuerdo con cualquiera de las reivindicaciones anteriores de la 22 a la 30, **caracterizado además porque** dicha unidad de textura (500) mapea las pinceladas en salidas del cálculo de iluminación dinámica.

15 **32.** El sistema de acuerdo con cualquiera de las reivindicaciones anteriores de la 22 a la 31, **caracterizado además porque** dicha coordenada de textura (500) puede estar generada negativamente para definir la retroiluminación de una imagen animada.

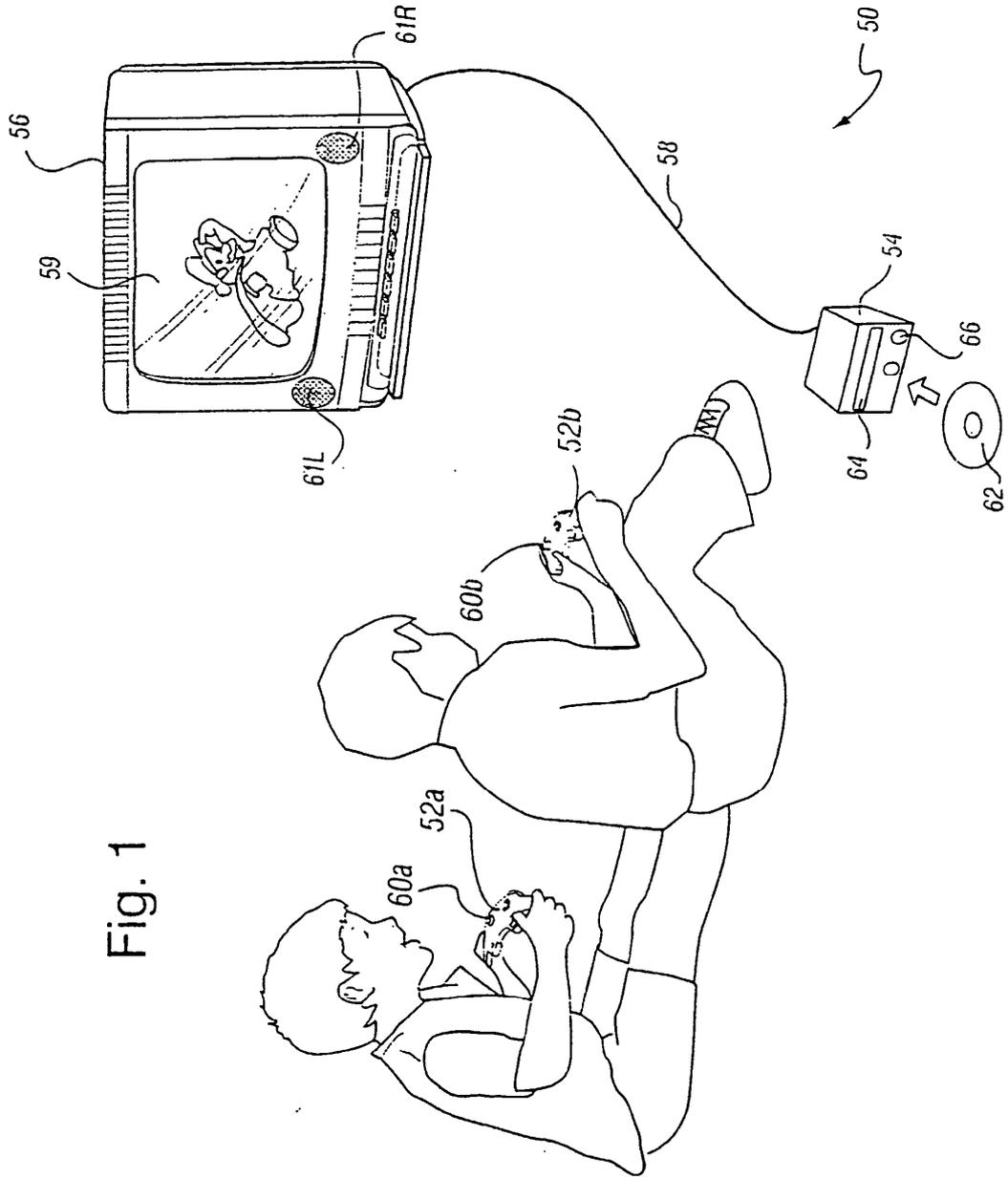
**33.** El método de acuerdo con cualquiera de las reivindicaciones anteriores de la 22 a la 32, **caracterizado porque** dicha salida de la tubería de datos de iluminación (300) define una primera coordenada de textura para dicho mapeado de textura y una segunda coordenada de textura para dicho mapeado de textura (500) se deriva de una fuente diferente a dicha salida de iluminación (302).

**34.** El sistema de acuerdo con cualquiera de las reivindicaciones anteriores de la 22 a la 33, **caracterizado además porque** dicha salida de la tubería de datos de iluminación (300) comprende una función de iluminación difusa.

25 **35.** El sistema de acuerdo con cualquiera de las reivindicaciones anteriores de la 22 a la 34, **caracterizado además porque** dicha salida de la tubería de datos de iluminación (300) comprende una función de iluminación especular.

**36.** El sistema de acuerdo con cualquiera de las reivindicaciones anteriores de la 22 a la 35, **caracterizado además porque** dicha salida de la tubería de datos de iluminación (300) comprende una función de iluminación de atenuación de distancia.

30 **37.** El sistema de acuerdo con cualquiera de las reivindicaciones anteriores de la 22 a la 36, **caracterizado además porque** dicha salida de iluminación comprende una función de iluminación de foco de luz.



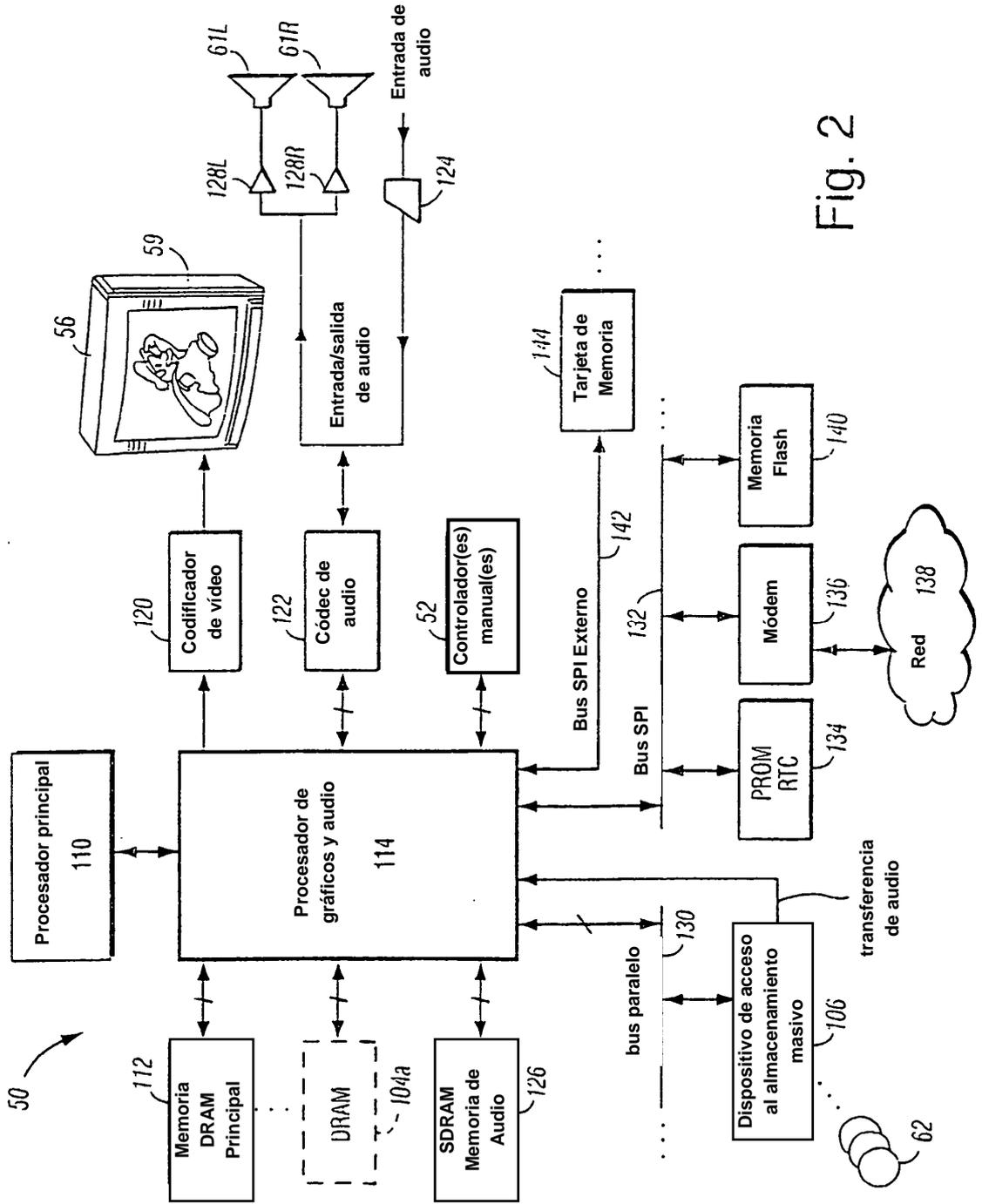


Fig. 2

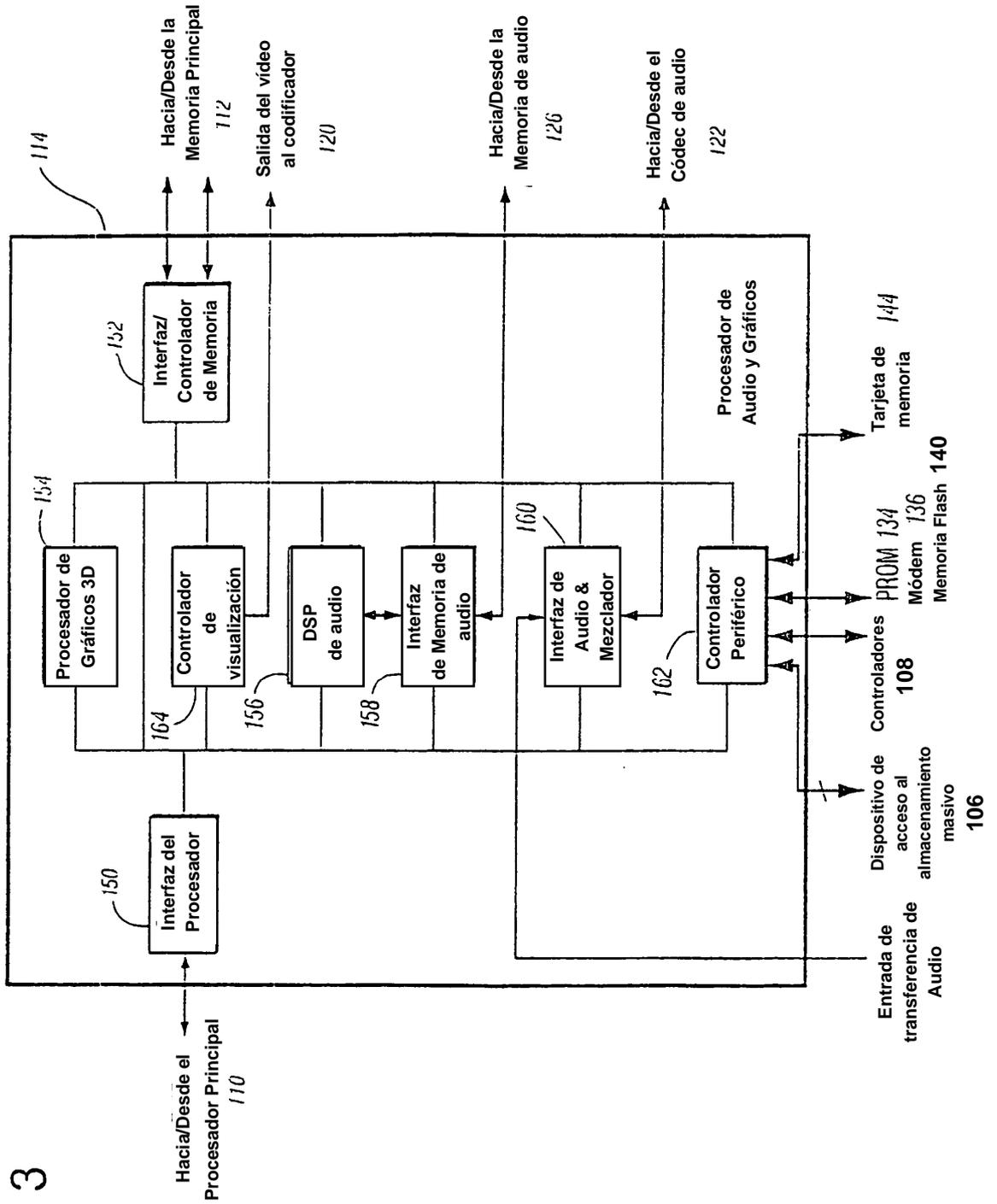


Fig. 3

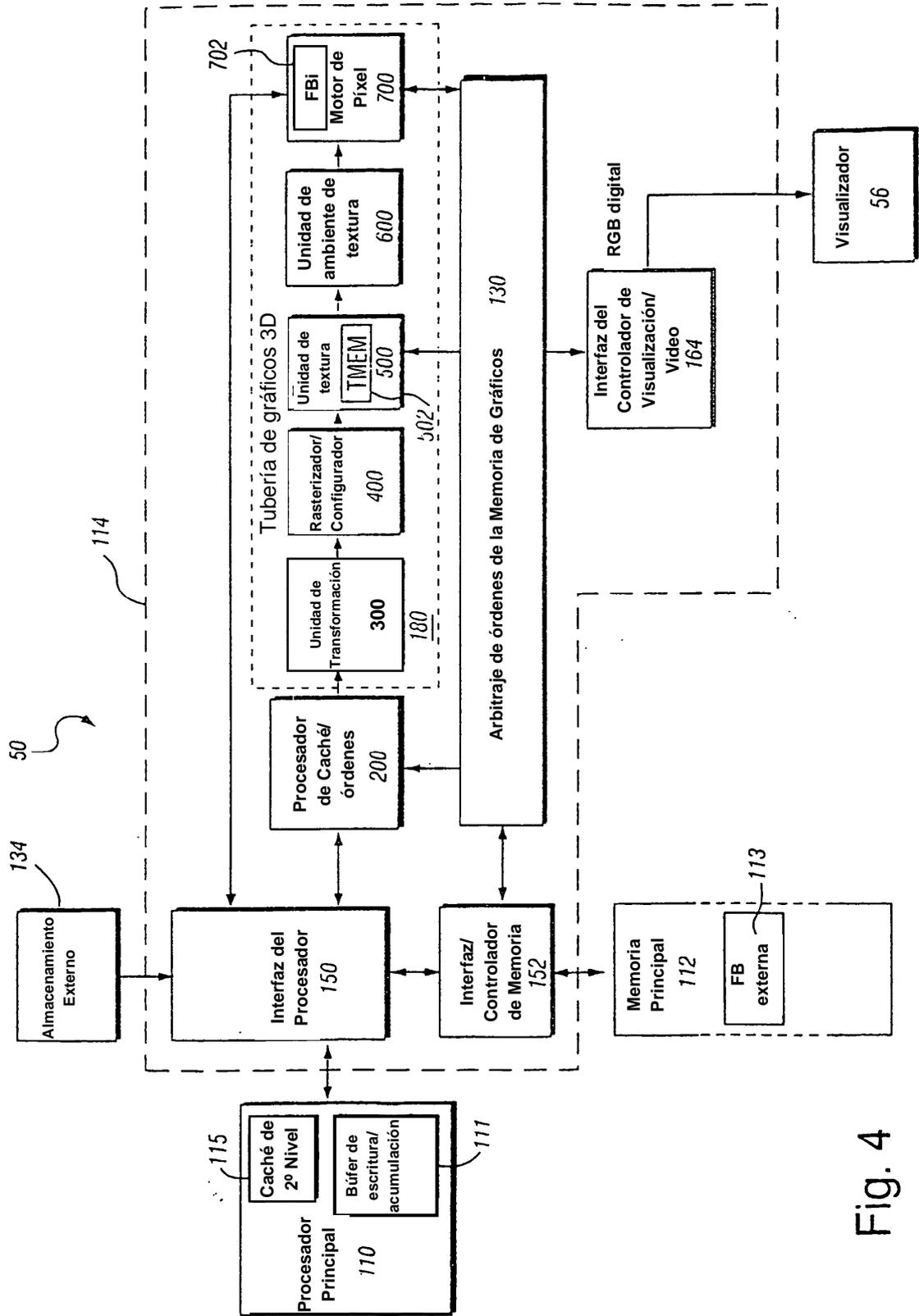


Fig. 4

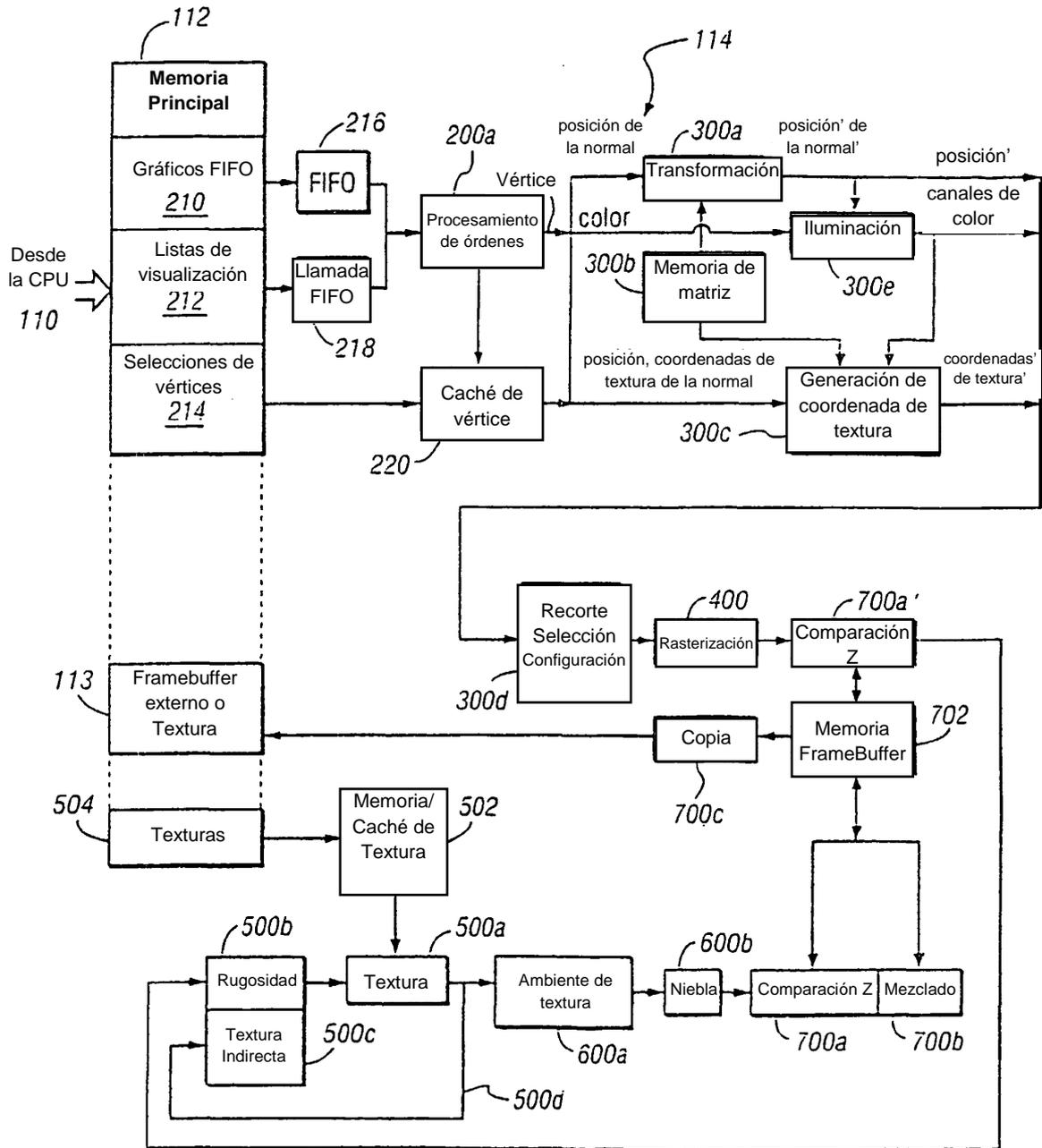


Fig. 5 EJEMPLO DE FLUJO DEL PROCESADOR DE GRÁFICOS

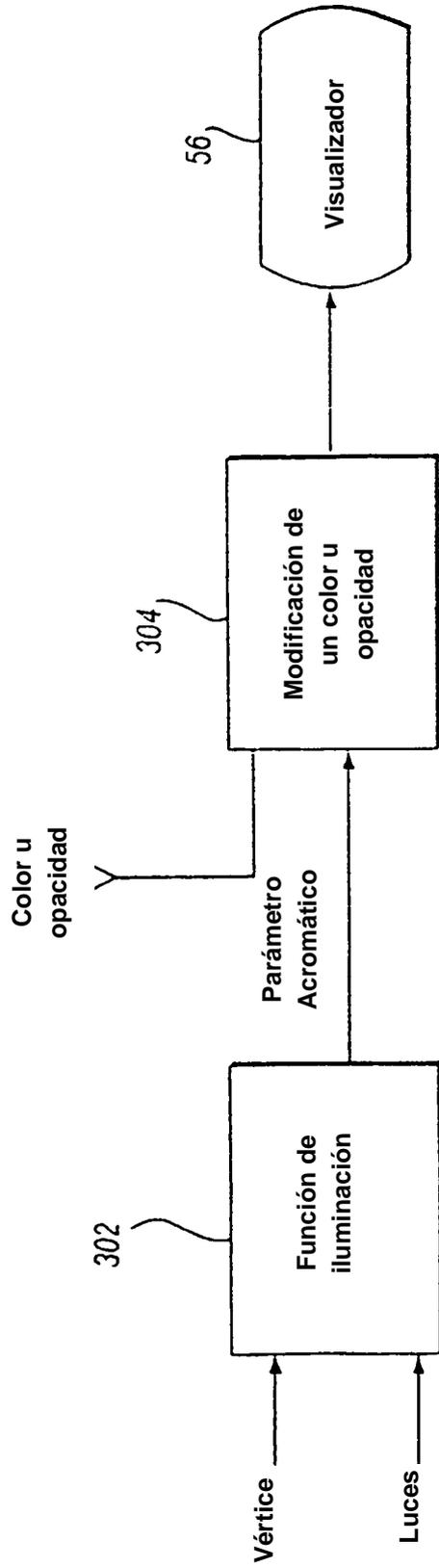


Fig. 6 EJEMPLO DEL PROCESO DE GENERACIÓN DE IMAGEN

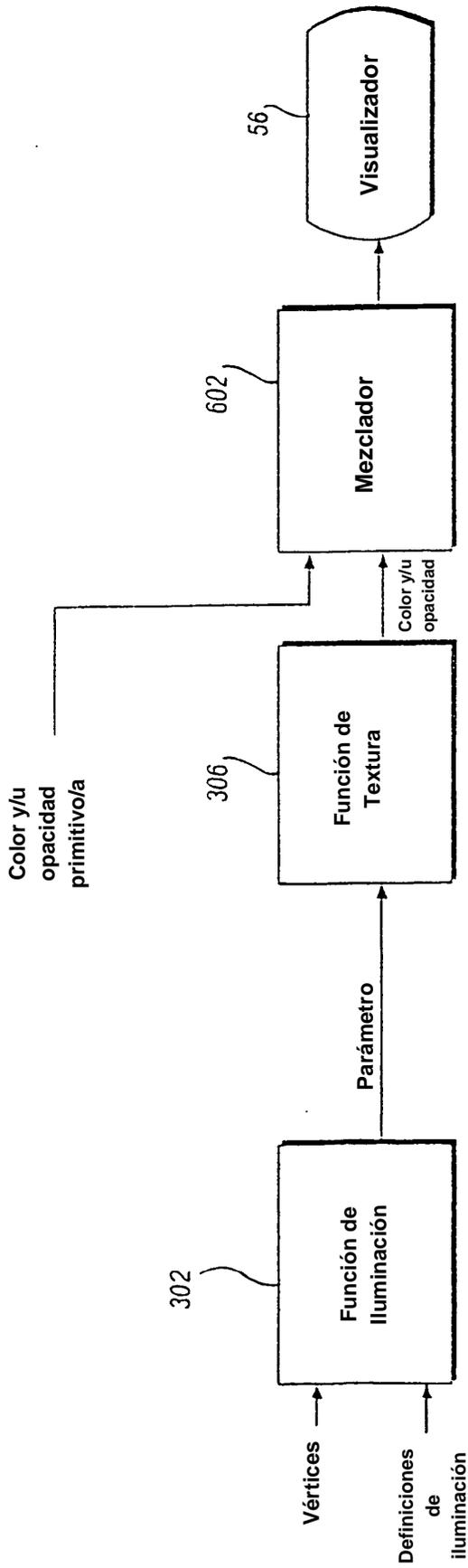


Fig. 7 EJEMPLO DEL PROCESO DE TEXTURIZACIÓN BASADO EN UN PARÁMETRO DE ILUMINACIÓN ACROMÁTICA

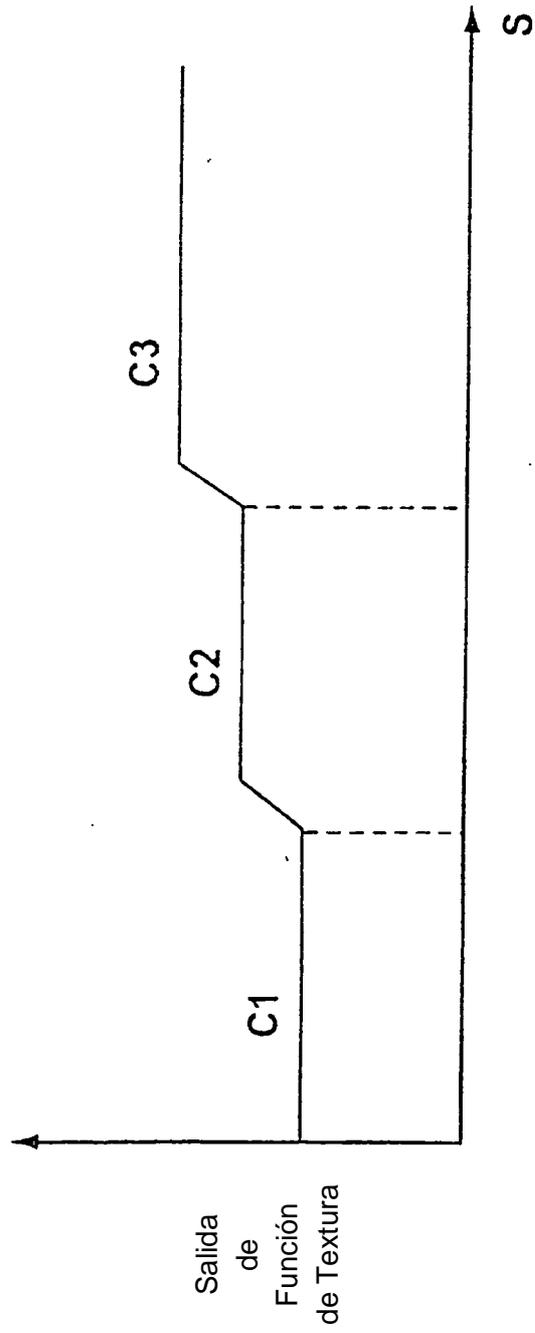


Fig. 8A EJEMPLO DE FUNCIÓN DE TEXTURA  
Salida = f(S)

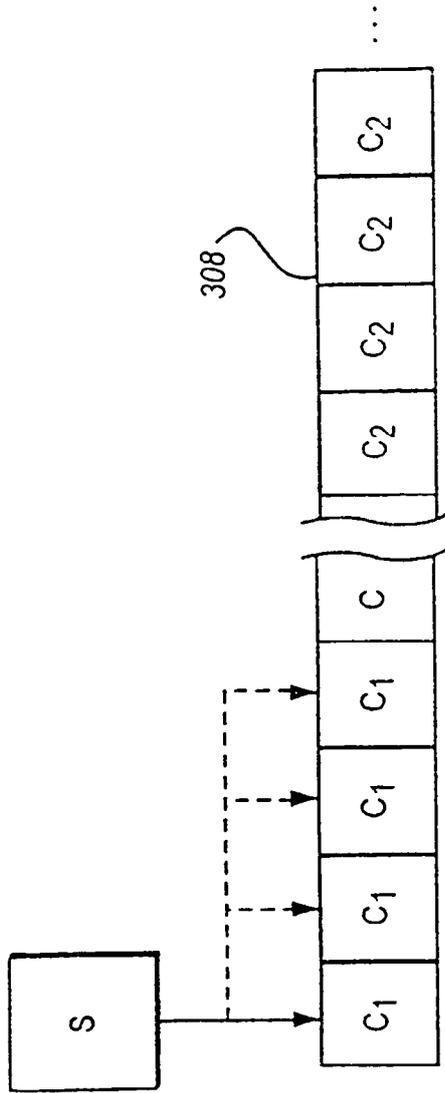


Fig. 8B EJEMPLO DE MAPEADO DE TEXTURA 1D

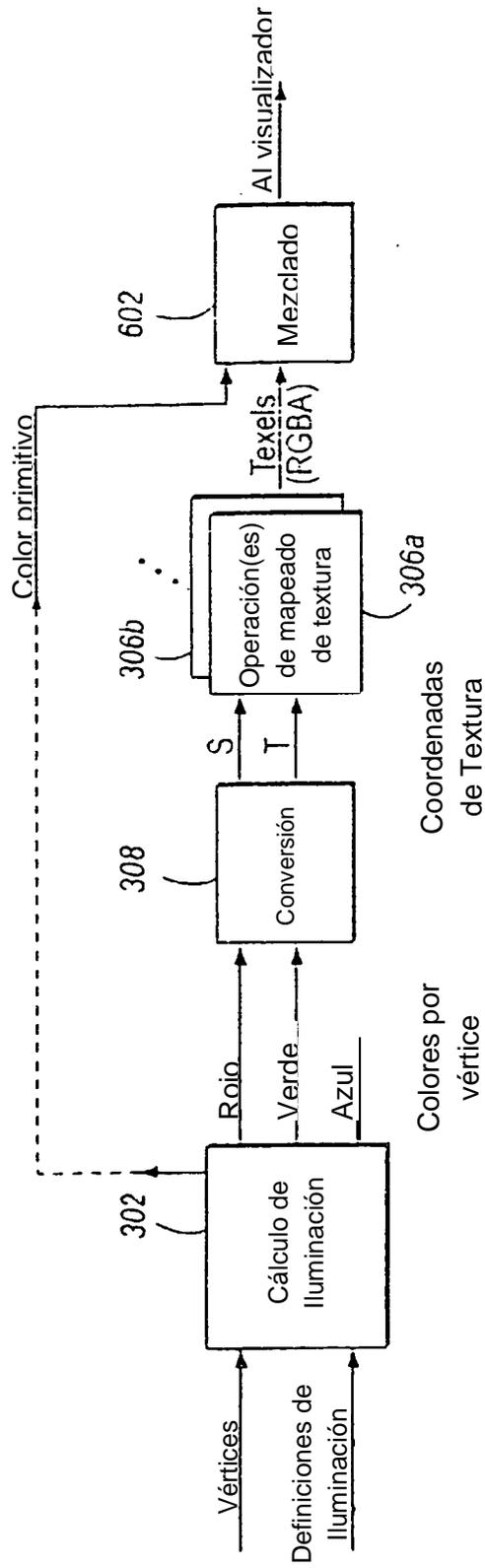


Fig. 9 EJEMPLO DE MAPEADO DE TEXTURA

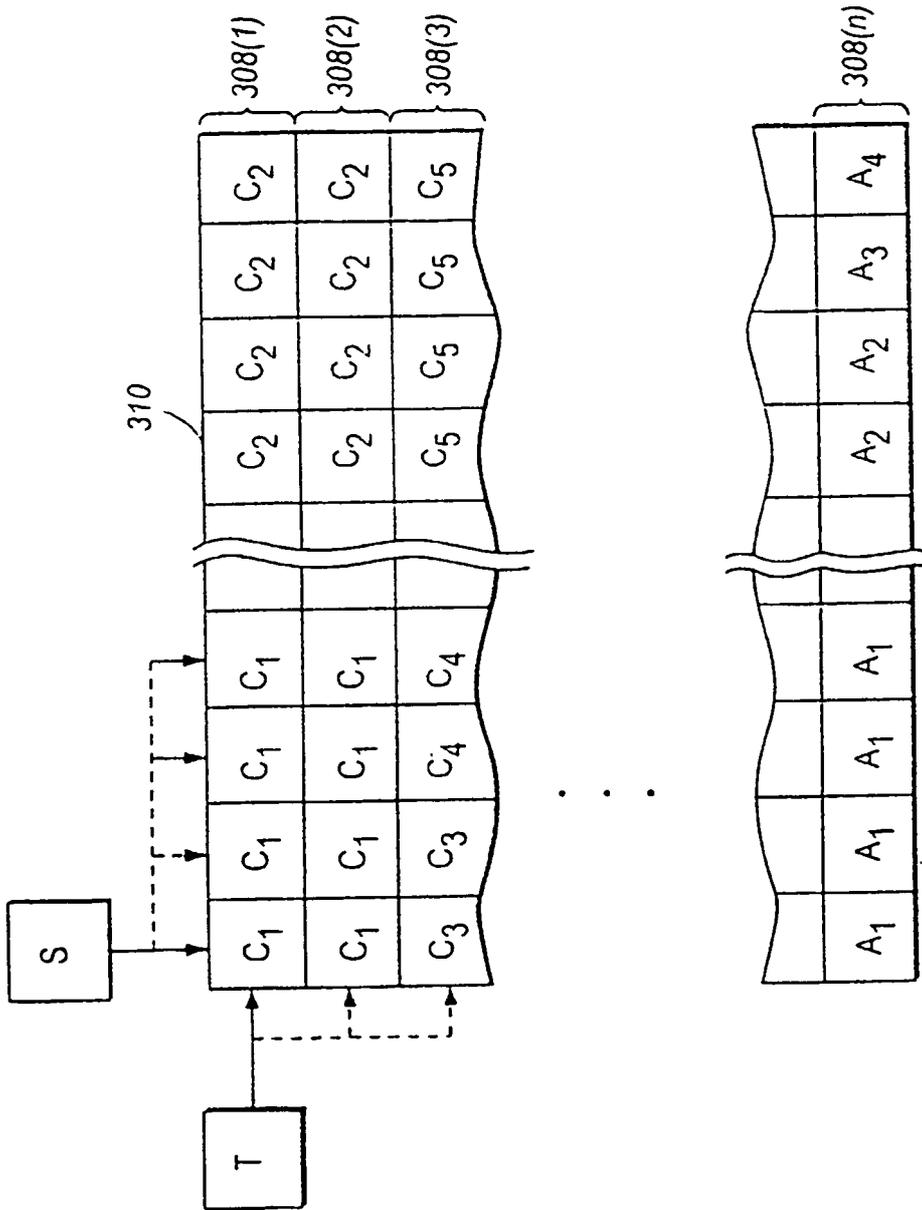


Fig. 10A EJEMPLO DE MAPEADO DE TEXTURA 2D

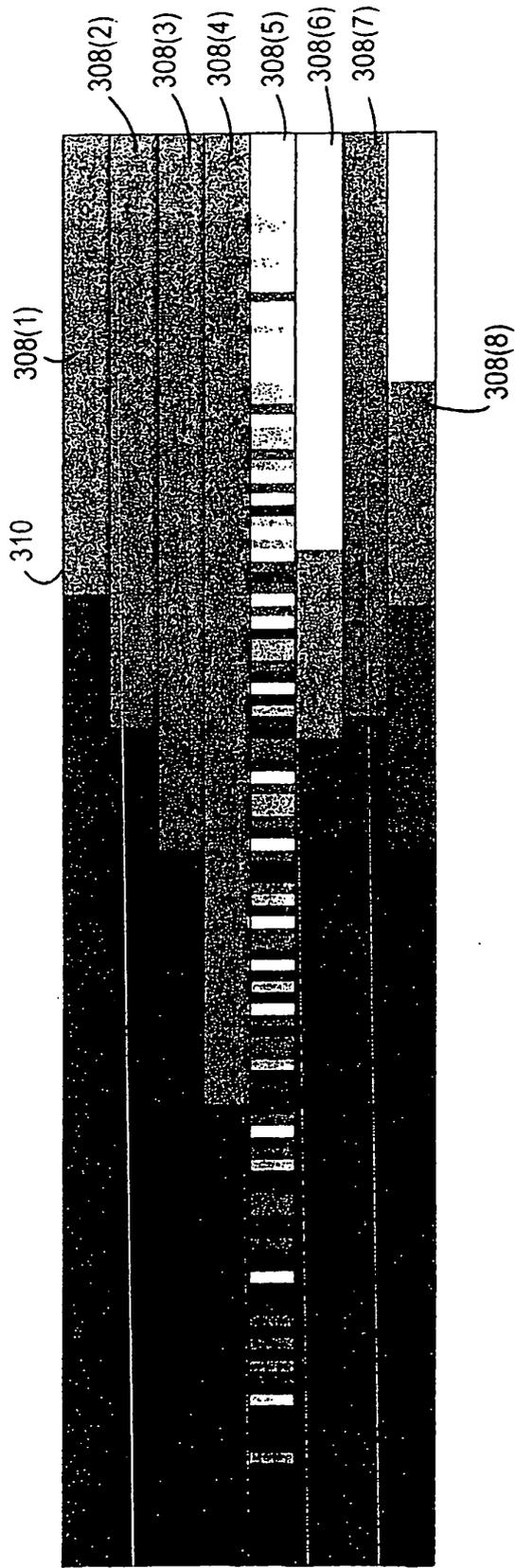


Fig. 10B

EJEMPLO DE TEXTURA 2D

Iluminación, Definiciones de vértice desde el Procesador de Órdenes 200

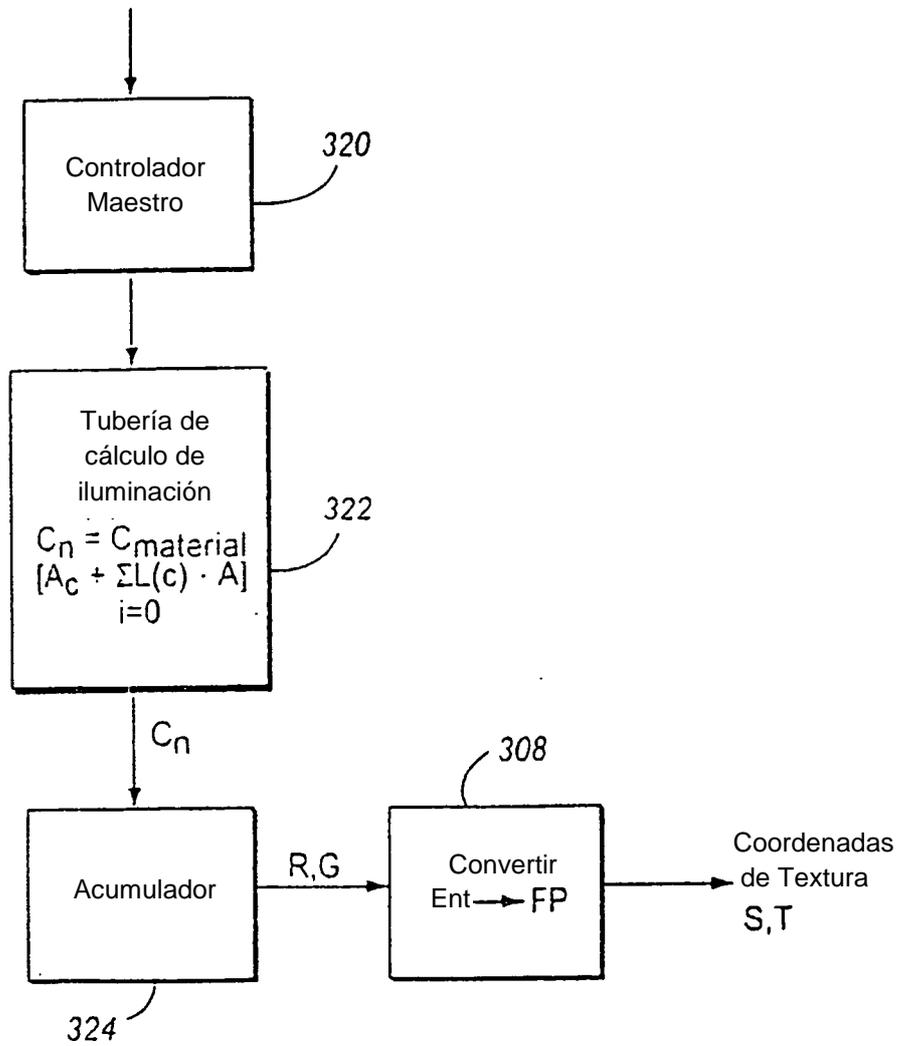


Fig. 11

EJEMPLO DE IMPLEMENTACIÓN DE LA TUBERÍA DE ILUMINACIÓN

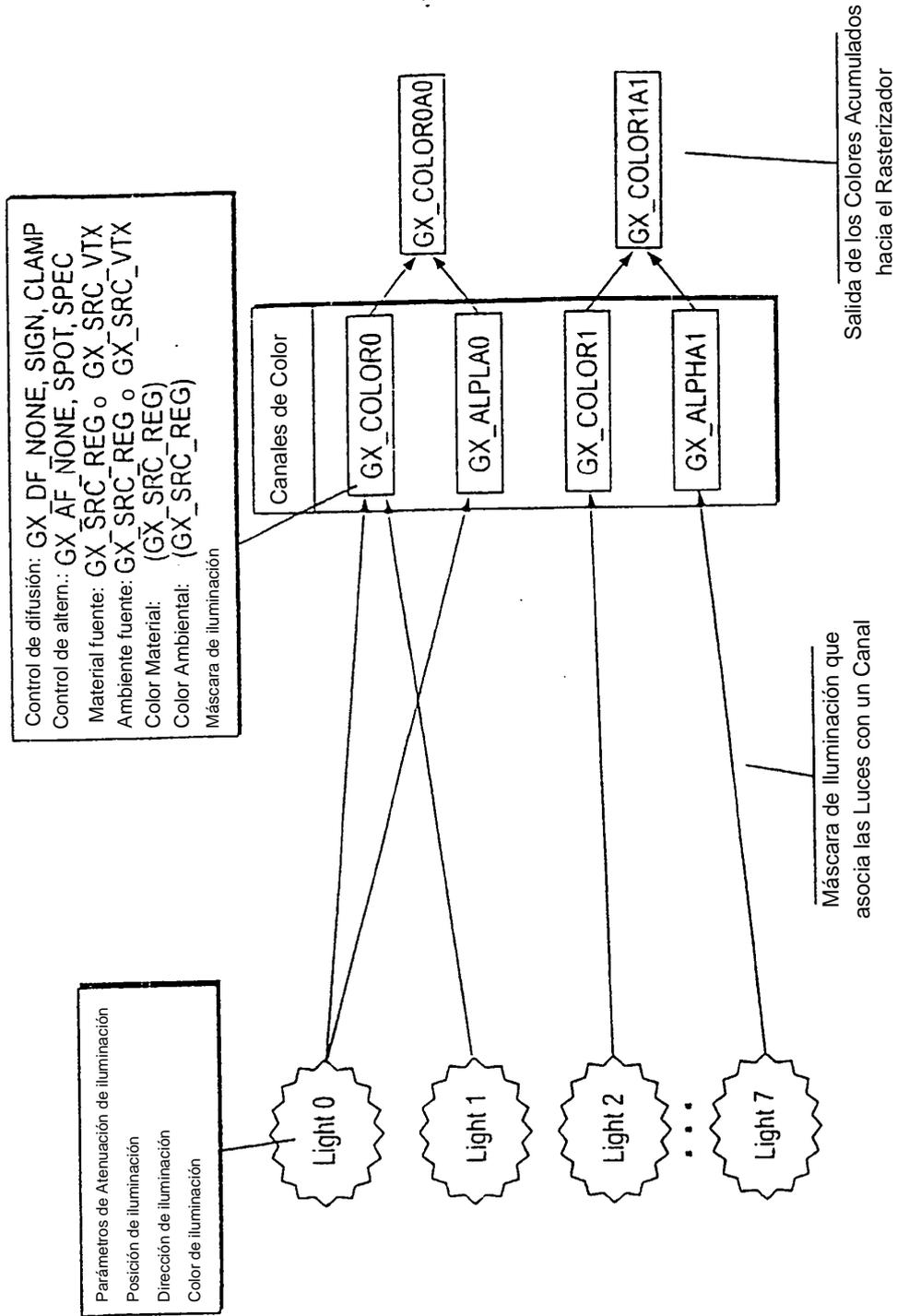


Fig. 12

ASOCIACIÓN DE LUCES CON CANALES DE COLOR

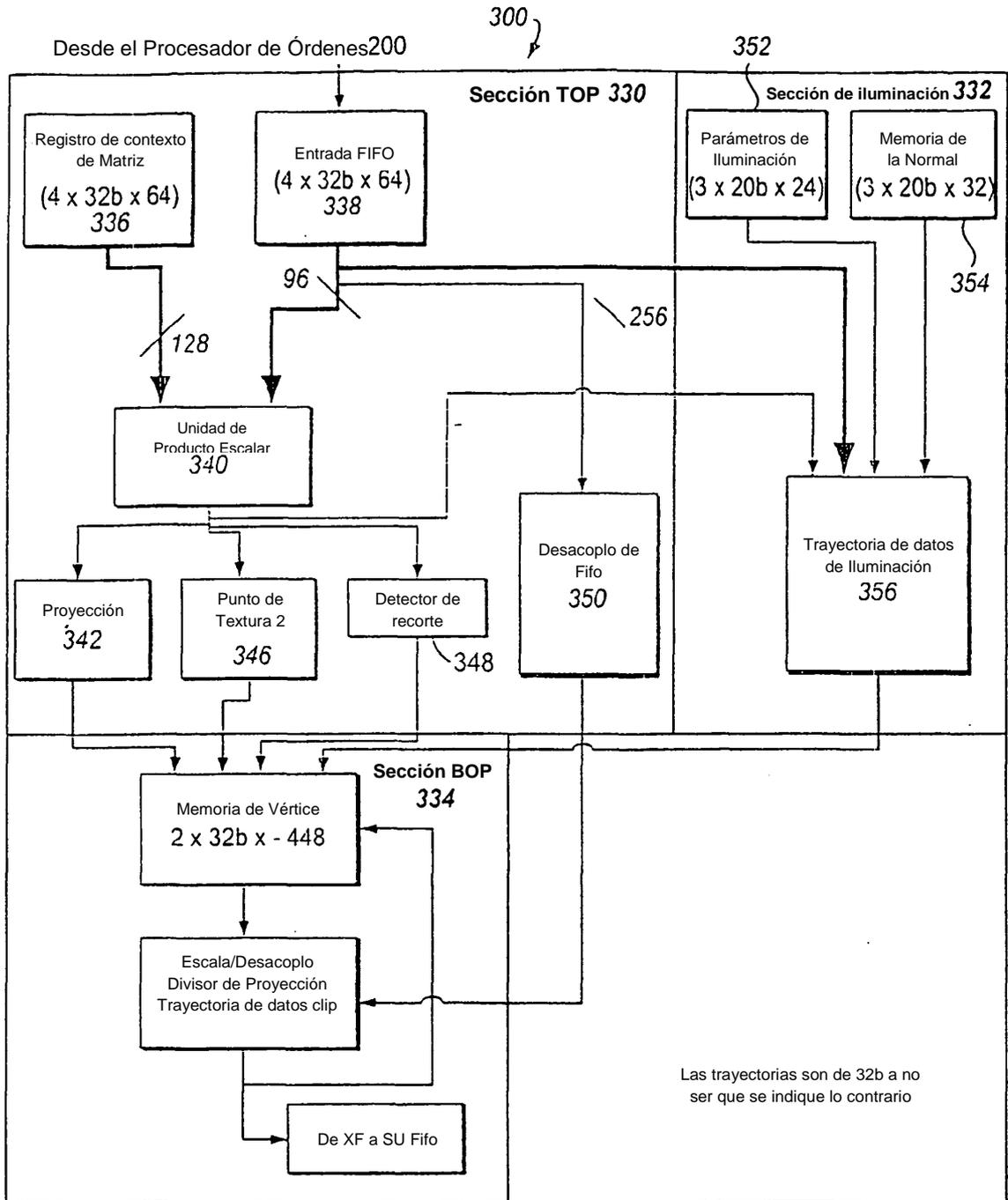


Fig. 13

EJEMPLO DE IMPLEMENTACIÓN DE LA UNIDAD DE TRANSFORMACIÓN

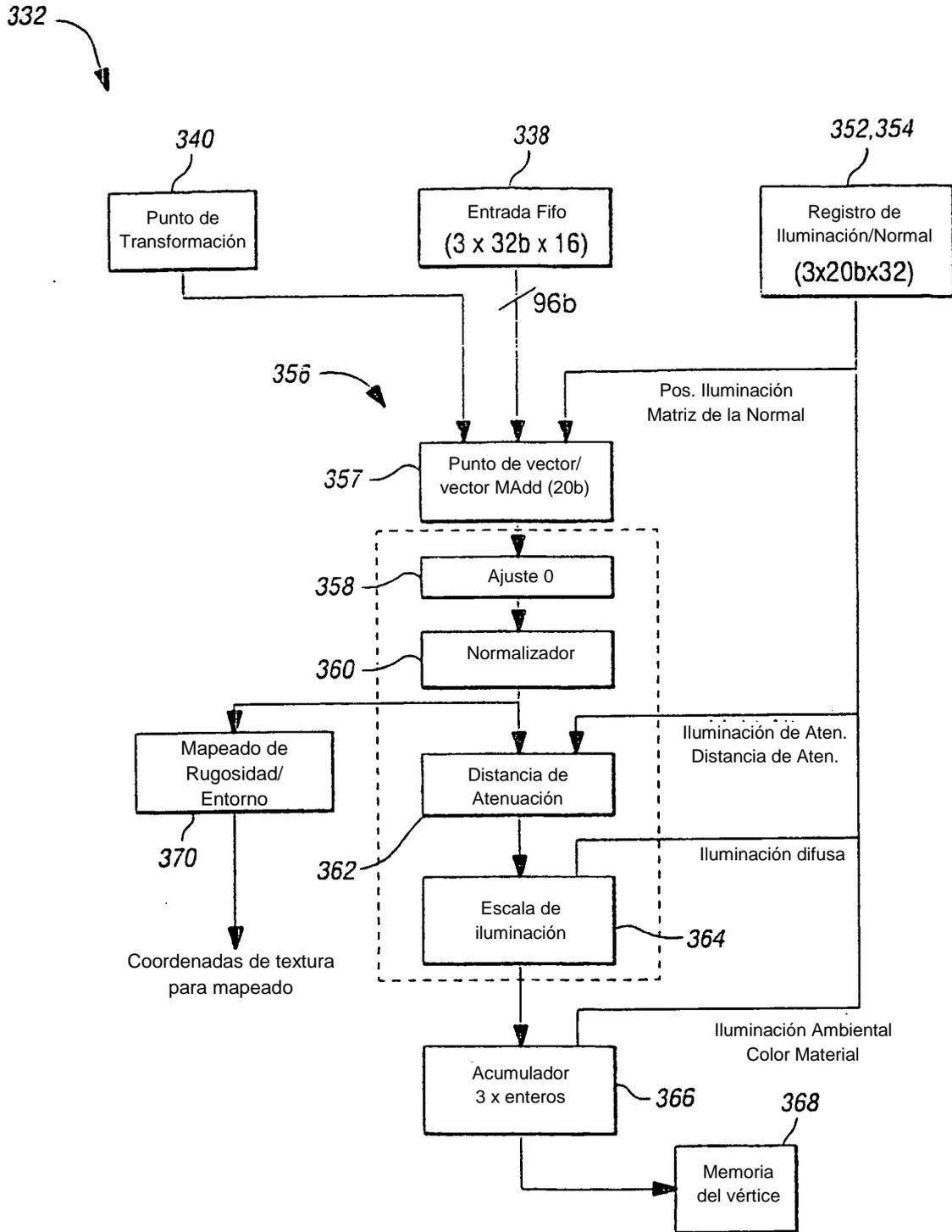


Fig. 14

Ejemplo de Implementación de la tubería de Iluminación

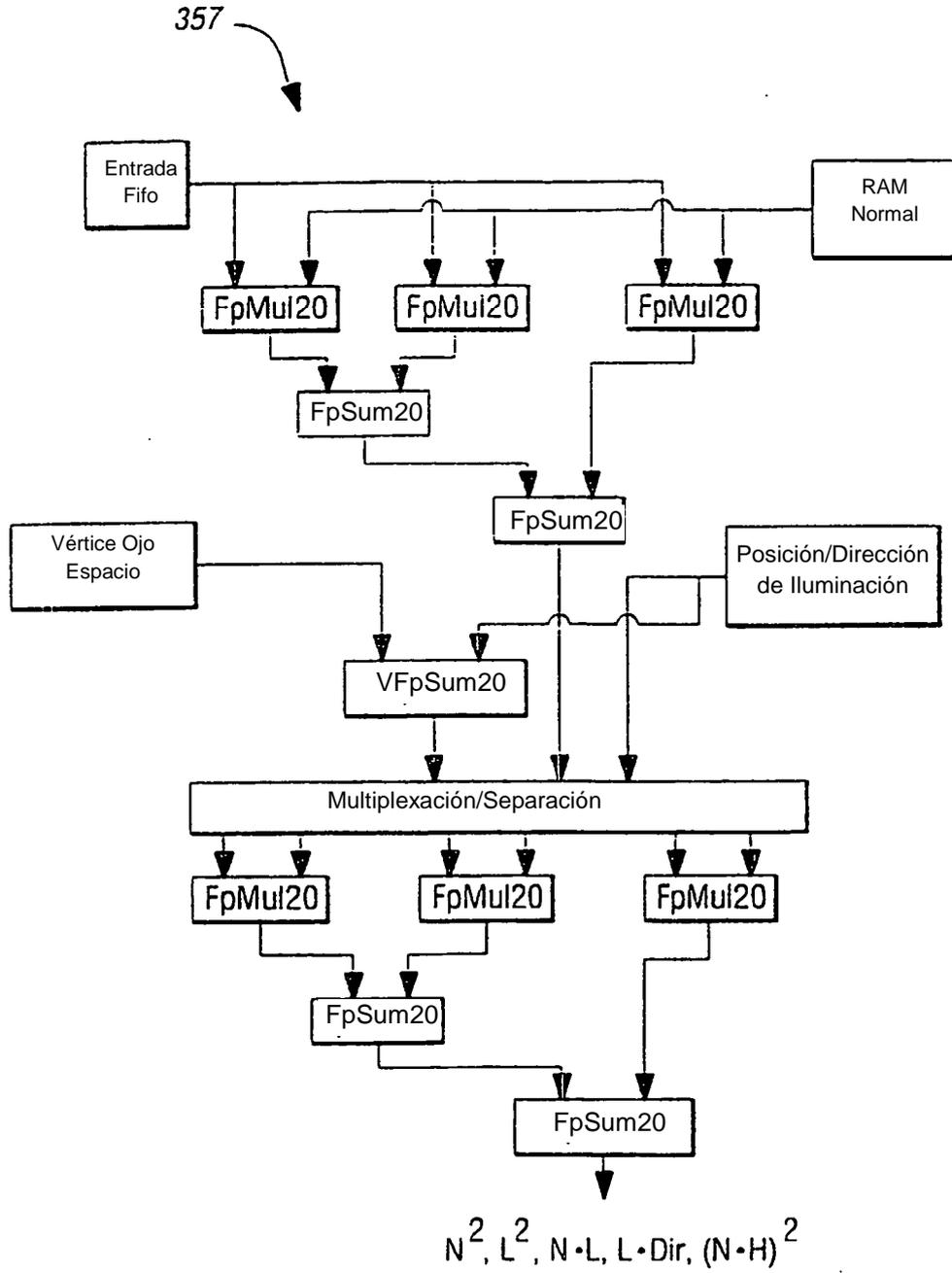


Fig. 15

Ejemplo de Unidad de Implementación de multiplicación/suma de vector

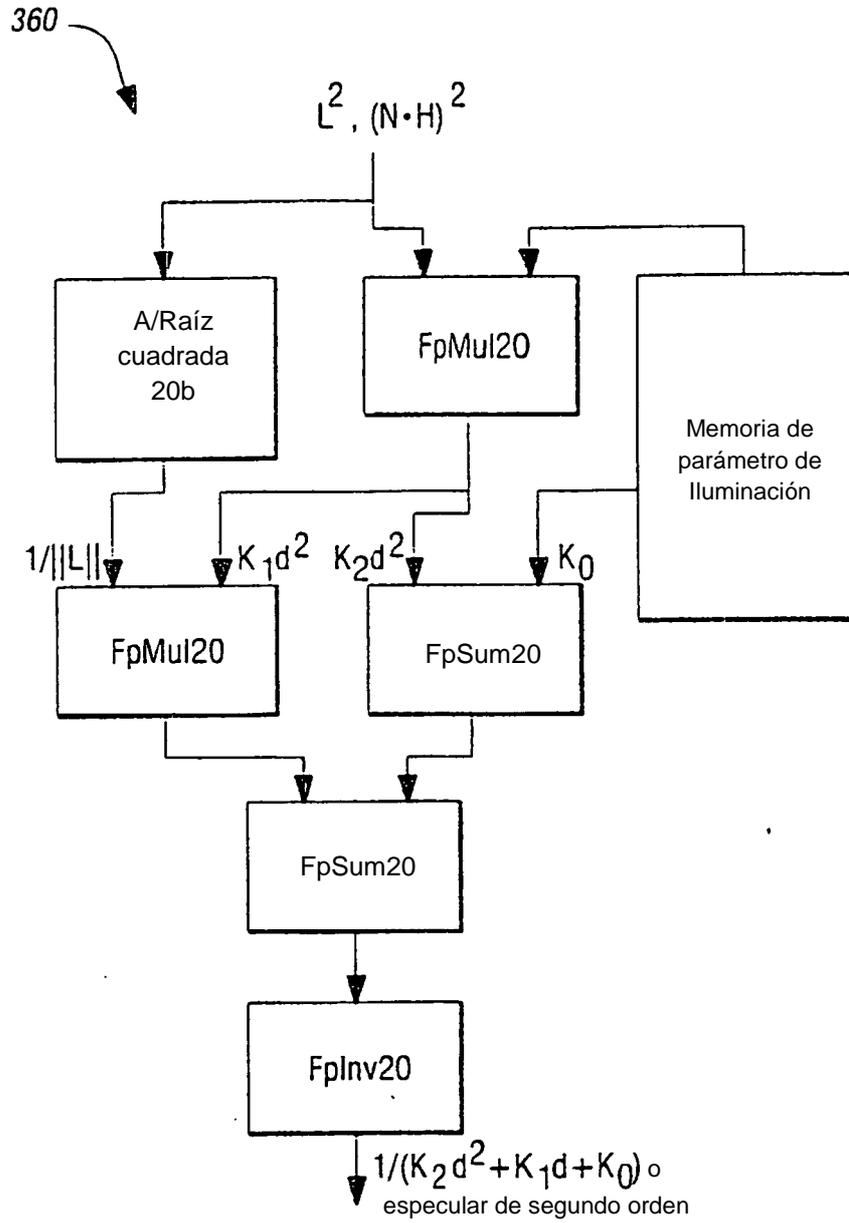


Fig. 16  
Ejemplo de Implementación de Normalización

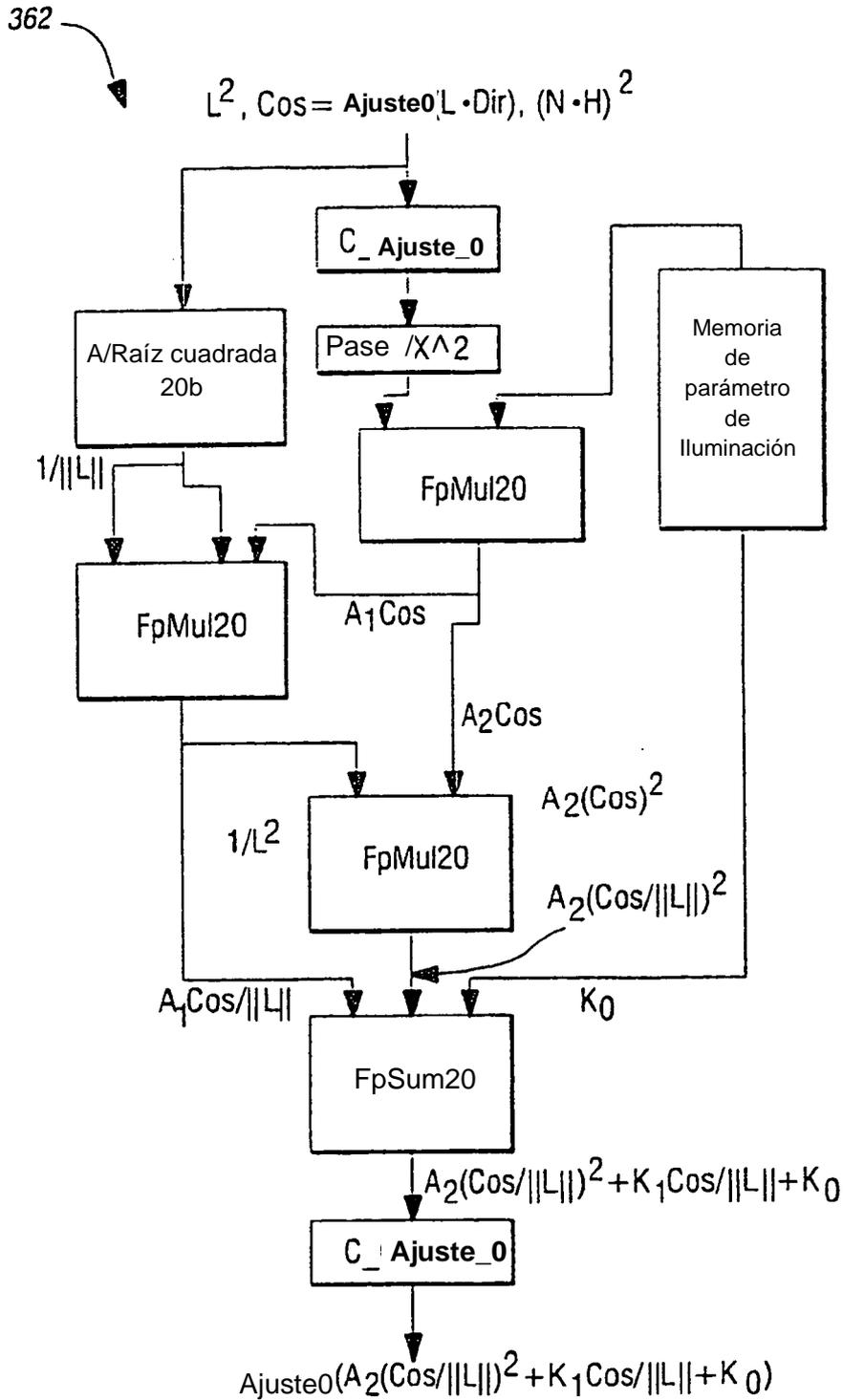
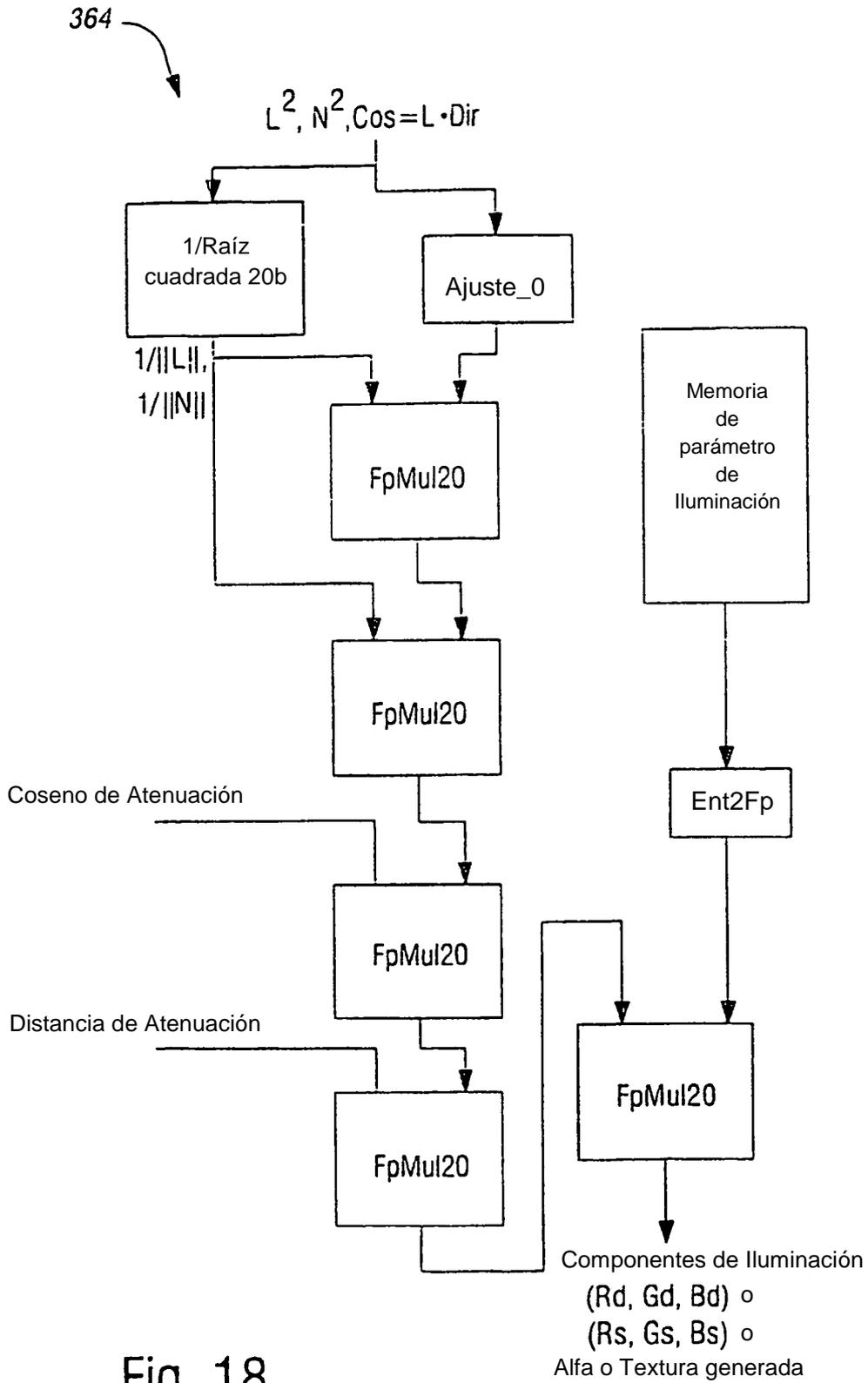


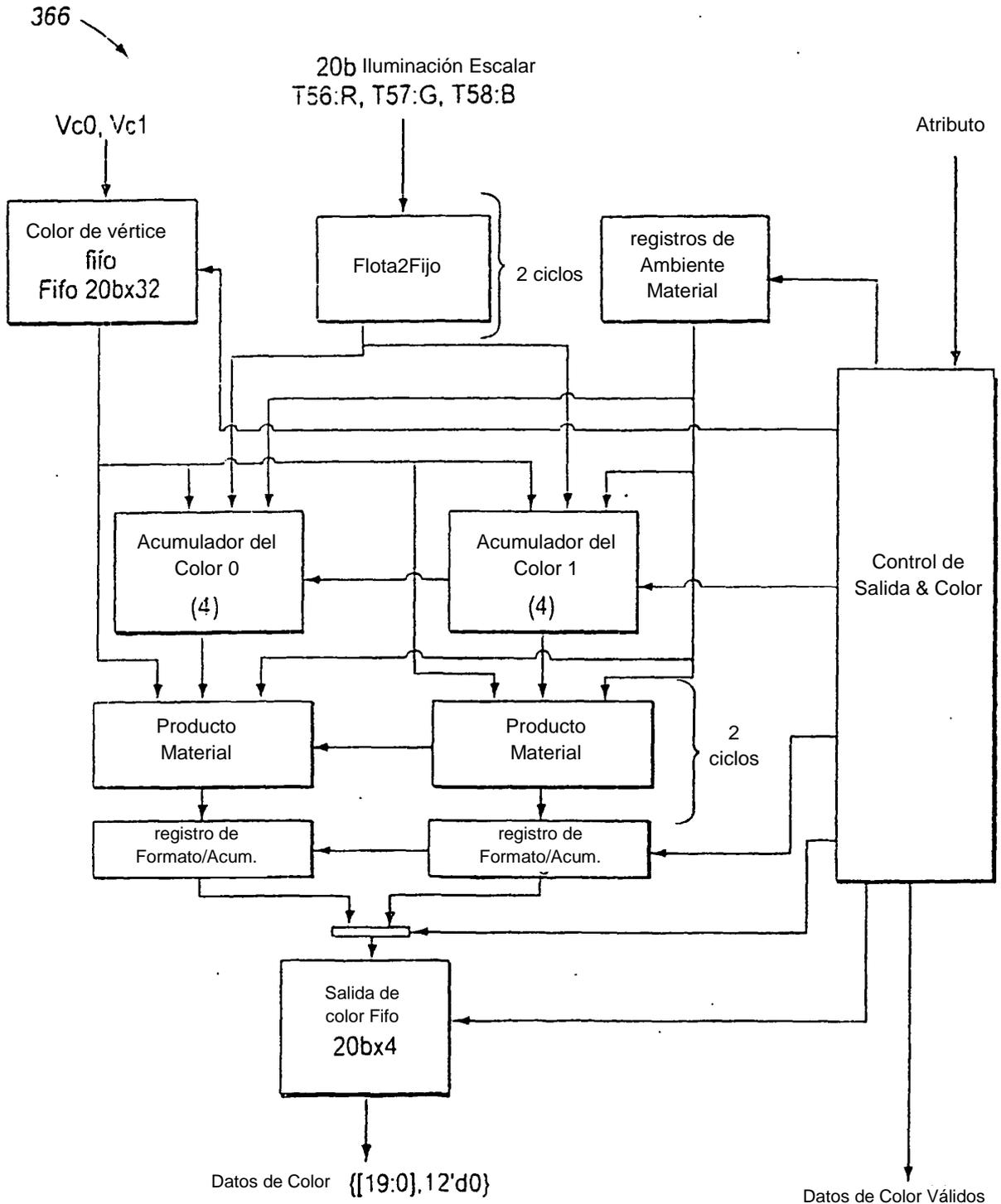
Fig. 17

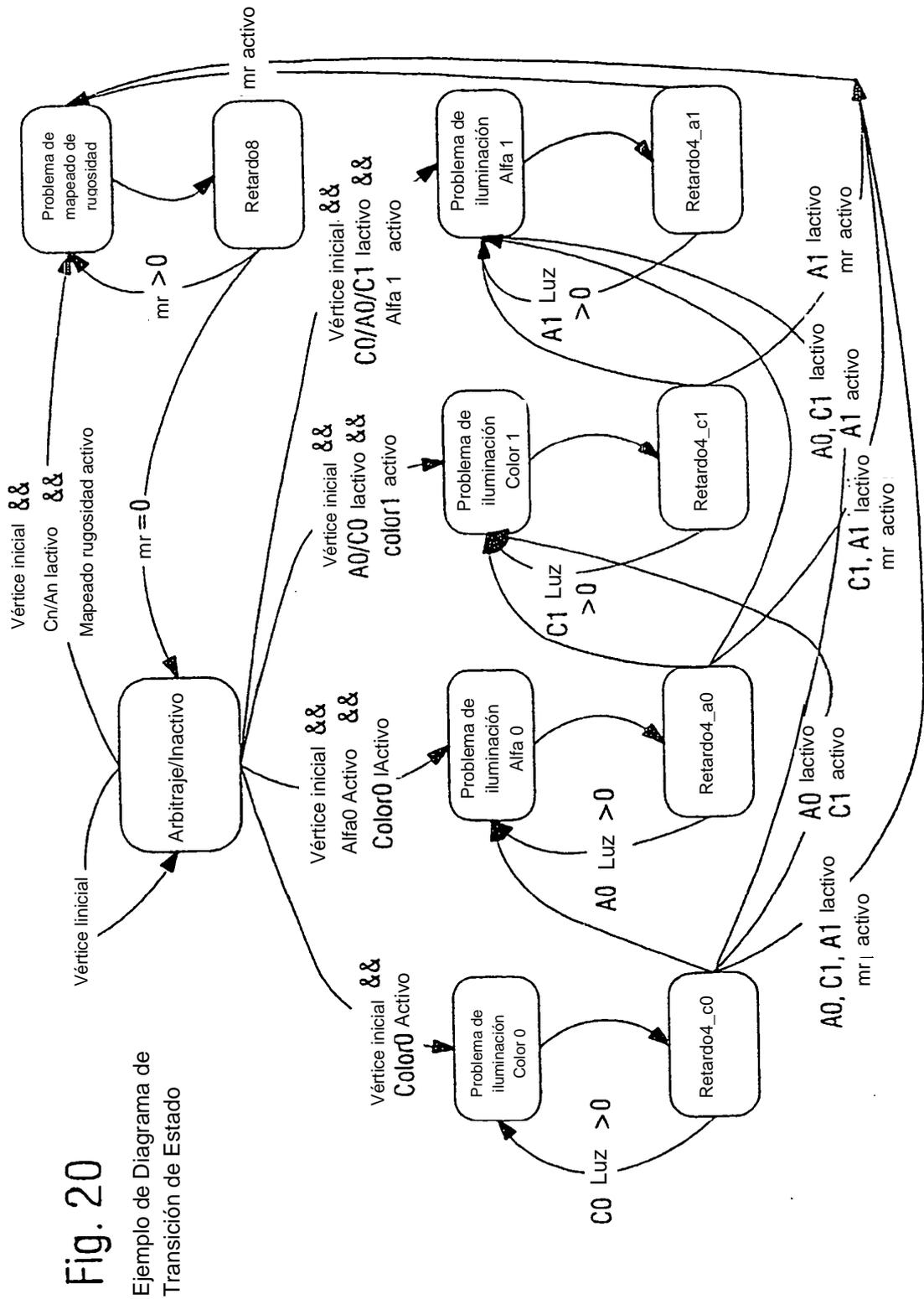
Ejemplo de Implementación Distancia de Atenuación



**Fig. 18**  
Ejemplo de Implementación  
de Escala de Iluminación

Fig. 19 EJEMPLO DE IMPLEMENTACIÓN DEL ACUMULADOR DE ENTEROS





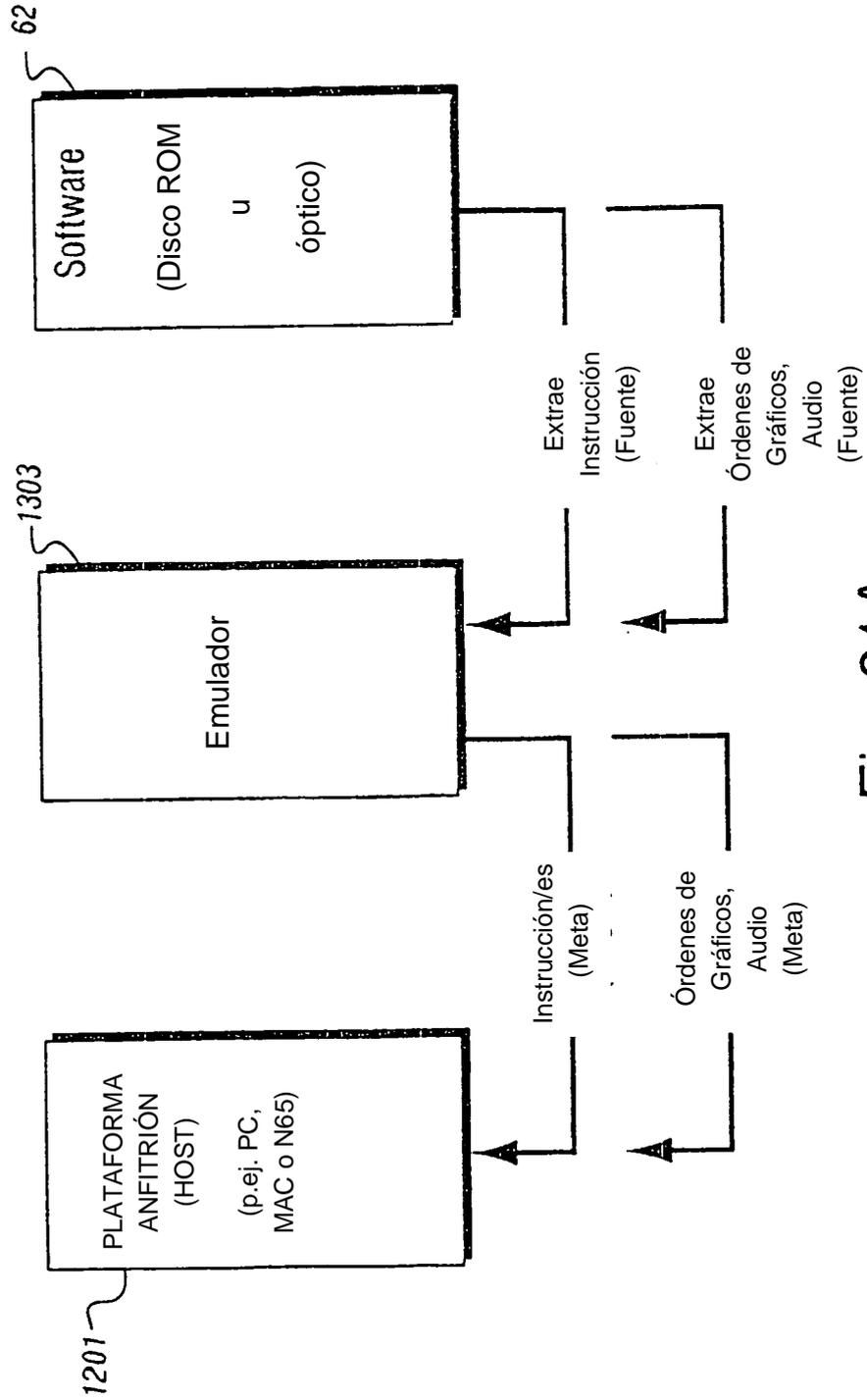


Fig. 21A

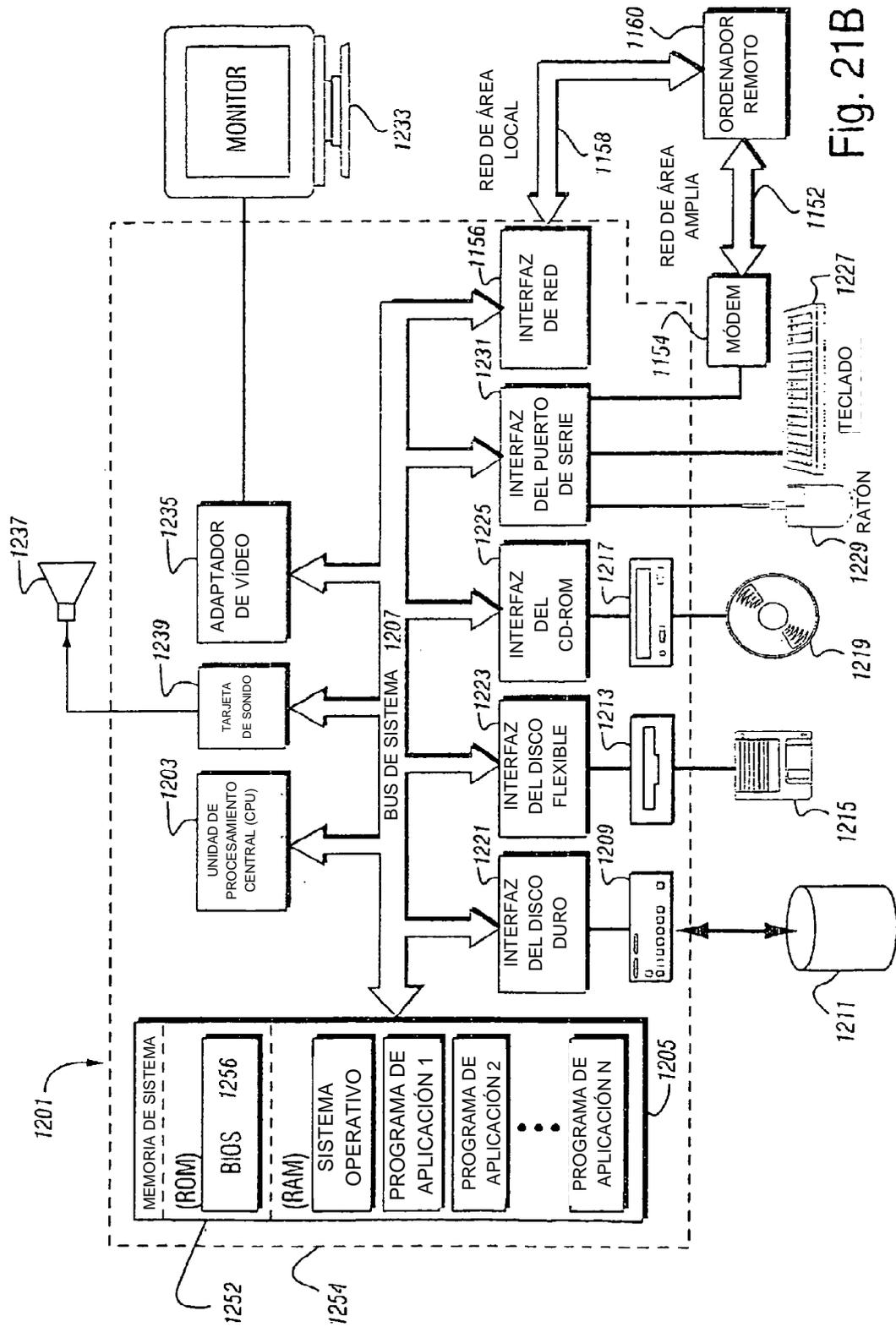


Fig. 21B