



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 282 136**

51 Int. Cl.:
G06T 5/00 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Número de solicitud europea: **00962075 .8**

86 Fecha de presentación : **08.09.2000**

87 Número de publicación de la solicitud: **1221139**

87 Fecha de publicación de la solicitud: **10.07.2002**

54

Título: **Procedimiento y aparato para producir una impresión de una imagen Bayer.**

30

Prioridad: **16.09.1999 AU PQ2890/99**

45

Fecha de publicación de la mención BOPI:
16.10.2007

45

Fecha de la publicación del folleto de la patente:
16.10.2007

73

Titular/es: **Silverbrook Research Pty. Limited**
393 Darling Street
Balmain, NSW 2041, AU

72

Inventor/es: **Lapstun, Paul y**
Walmsley, Simon Robert

74

Agente: **Carpintero López, Francisco**

ES 2 282 136 T3

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

DESCRIPCIÓN

Procedimiento y aparato para producir una impresión de una imagen Bayer.

5 **Campo de la invención**

La presente invención se refiere a un procedimiento y a un aparato para producir una impresión de una imagen Bayer.

10 La invención ha sido desarrollada principalmente para una cámara digital que incluye una impresora integral para proporcionar una impresión sobre papel de una imagen capturada por la cámara, y va a ser descrita en lo que sigue con referencia a esa aplicación. Sin embargo, se apreciará que la invención no se limita a ese campo de uso particular.

Sumario de la invención

15 De acuerdo con un primer aspecto de la invención, se proporciona un procedimiento para la provisión de una imagen para su impresión a una resolución de puntos bi-nivel predeterminada, que corresponde con una resolución de tono continuo predeterminada, incluyendo el procedimiento las etapas de:

20 recibir un primer conjunto de datos indicativos de la imagen, estando el primer conjunto de datos en formato Bayer por planos, que comprende diferentes planos (45, 46, 47) de colores, siendo al menos uno de los planos (46) de un color de una resolución diferente a la del plano o planos (45, 47) de otro color,

25 realizar una reconstrucción y un remuestreo (64) de la imagen de cada plano de color del primer conjunto de datos para crear un segundo conjunto de datos de resolución de tono continuo predeterminada, que sea mayor que la resolución de cada uno de los planos de color del primer conjunto de datos;

convertir (67) el segundo conjunto de datos en un tercer conjunto de datos de resolución de puntos bi-nivel predeterminada, que sea mayor que la resolución de tono continuo predeterminada del segundo conjunto de datos, y

30 hacer que el tercer conjunto de datos se encuentre disponible para una impresora (69) a la resolución predeterminada de puntos bi-nivel.

35 De preferencia, la primera resolución equivale a la resolución de puntos bi-nivel predeterminada. En otras realizaciones, sin embargo, la primera resolución es mayor que la resolución de puntos bi-nivel predeterminada. Todavía en otra realización, la primera resolución es menor que la resolución de puntos bi-nivel predeterminada.

40 También de preferencia, el primer conjunto de datos está en formato rojo, verde y azul RGB, y la impresora actúa en respuesta a un formato cyan, magenta y amarillo CMY, y el procedimiento incluye la etapa adicional de convertir el tercer conjunto de datos desde el formato RGB al formato CMY.

En una forma preferida, el procedimiento incluye la etapa de hacer más intenso el segundo conjunto de datos. Alternativamente, el procedimiento incluye la etapa preferida de hacer más intenso el primer conjunto de datos.

45 De preferencia, el primer conjunto de datos se obtiene a partir de un dispositivo sensor, y el procedimiento incluye la etapa de compensar el primer conjunto de datos en cuanto a las no linealidades del dispositivo sensor. Más preferentemente, la etapa de compensar incluye convertir el primer conjunto de datos desde una pluralidad de muestras de bits x hasta una pluralidad de muestras de bits y , donde $x > y$. Incluso de manera más preferente, $x = 10$ e $y = 8$.

50 También de preferencia, el procedimiento incluye la etapa de disponer en planos el primer conjunto de datos mediante un plano rojo, un plano verde y un plano azul.

En una forma preferida, el procedimiento incluye las etapas adicionales de: determinar para el primer conjunto de datos, el $m\%$ de píxeles más oscuros y el $n\%$ de píxeles más brillantes;

55 ajustar el primer conjunto de datos para igualar el $m\%$ de píxeles más oscuros, y

ajustar el primer conjunto de datos para igualar el $n\%$ de píxeles más brillantes.

60 De preferencia, el procedimiento incluye la etapa adicional de ajustar el primer conjunto de datos para proporcionar un balance del blanco predeterminado. Más preferiblemente, el procedimiento incluye la etapa adicional de ajustar el primer conjunto de datos para proporcionar una expansión de rango predeterminada. Incluso más preferiblemente, la resolución de color del primer conjunto de datos se incrementa mientras se mantiene la misma resolución espacial.

65 En una forma preferida, el primer conjunto de datos se ajusta selectivamente para proporcionar la imagen con una orientación rotacional predeterminada.

ES 2 282 136 T3

De acuerdo con un segundo aspecto de la invención, se proporciona un aparato para la provisión de una imagen, para su impresión a una resolución predeterminada de puntos bi-nivel que corresponde con una resolución predeterminada de tono continuo, incluyendo el aparato:

5 medios de entrada para recibir un primer conjunto de datos indicativo de la imagen, estando el primer conjunto de datos en formato Bayer por planos, comprendiendo planos (45, 46, 47) de diferentes colores siendo al menos uno de los planos (46) de color de una resolución diferente a la del (de los) otro(s) plano o planos (45, 47) de color,

10 medios de muestreo para reconstrucción y remuestreo (64) de imagen en el primer conjunto de datos, para crear un segundo conjunto de datos con una resolución de tono continuo predeterminada, que es mayor que la resolución de cada uno de los planos de color del primer conjunto de datos;

15 medios de procesamiento para convertir (67) el segundo conjunto de datos en un tercer conjunto de datos con una resolución predeterminada de puntos bi-nivel que es mayor que la resolución predeterminada de tono continuo del segundo conjunto de datos, y

20 medios de salida para hacer que el tercer conjunto de datos esté disponible (69) para su impresión a la resolución predeterminada de puntos bi-nivel.

De preferencia, la primera resolución se equipara con la resolución predeterminada de puntos bi-nivel. Alternativamente, la primera resolución es mayor que la resolución predeterminada de puntos bi-nivel. En otra realización, sin embargo, la primera resolución es menor que la resolución predeterminada de puntos bi-nivel.

25 También de preferencia, el primer conjunto de datos está en un formato rojo, verde y azul (RGB), y la impresora actúa en respuesta a un formato cyan, magenta y amarillo (CMY), en el que los medios de procesamiento convierten el tercer conjunto de datos desde el formato RGB al formato CMY.

30 También de preferencia, el aparato hace más intenso el segundo conjunto de datos. En otra realización, sin embargo, el aparato intensifica el primer conjunto de datos.

35 En una forma preferida, el primer conjunto de datos se obtiene a partir de un dispositivo sensor, y los medios de entrada compensan el primer conjunto de datos en cuanto a las no linealidades del dispositivo sensor. Más preferiblemente, la compensación para las no linealidades incluye convertir el primer conjunto de datos, desde una pluralidad de muestras de bits x , a una pluralidad de muestras de bits y , donde $x > y$. Incluso más de preferencia, $x = 10$, e $y = 8$.

40 De preferencia, los medios de entrada disponen en planos el primer conjunto de datos según un plano rojo, un plano verde y un plano azul.

Más preferiblemente, los medios de entrada:

determinan, para el primer conjunto de datos, el $m\%$ de píxeles más oscuros y el $n\%$ de píxeles más brillantes;

45 ajustan el primer conjunto de datos para igualar el $m\%$ de píxeles más oscuros, y

ajustan el primer conjunto de datos para igualar el $n\%$ de píxeles más brillantes.

50 En una forma preferida, los medios de entrada ajustan el primer conjunto de datos para proporcionar un balance del blanco predeterminado. Más preferiblemente, los medios de entrada ajustan el primer conjunto de datos para proporcionar una expansión de rango predeterminada. Incluso más preferiblemente, los medios de entrada incrementan la resolución de color del primer conjunto de datos mientras mantienen la misma resolución espacial.

55 De preferencia, los medios de entrada ajustan selectivamente el primer conjunto de datos para proporcionar la imagen con una orientación rotacional predeterminada.

De acuerdo con un tercer aspecto de la invención, se proporciona una cámara que incluye:

60 un dispositivo sensor CCD para la provisión de una imagen Bayer;

una impresora para proporcionar selectivamente una imagen impresa, y

65 un aparato según se ha descrito anteriormente, para recibir la imagen Bayer y proporcionar a la impresora el tercer conjunto de datos de tal modo que se pueda producir la imagen impresa.

ES 2 282 136 T3

Breve descripción de los dibujos

Las realizaciones preferidas de la invención van a ser descritas ahora, únicamente a título de ejemplo, con referencia a la descripción y a las Figuras que siguen.

- 5 La Figura 1 muestra un flujo de imagen de alto nivel del PCP;
la Figura 2 muestra un diagrama del PCP por separado;
- 10 la Figura 3 muestra un diagrama de bloques del PCP conectado al hardware de la Printcam;
la Figura 4 muestra una cabeza de impresión Memjet de 10 cm (4 pulgadas);
la Figura 5 muestra la disposición de segmentos en una cabeza de impresión de 10 cm (4 pulgadas);
- 15 la Figura 6 muestra la disposición de boquillas en un pod, numeradas por orden de disparo;
la Figura 7 muestra la disposición de boquillas en un pod, numeradas por orden de carga;
- 20 la Figura 8 muestra un cromo-pod;
la Figura 9 muestra un grupo-pod;
la Figura 10 muestra un grupo de fase;
- 25 la Figura 11 muestra la relación entre segmentos, grupos de disparo, grupos de fase, grupo-pods y cromo-pods;
la Figura 12 muestra perfiles de pulso de AEnable y BEnable durante la impresión de un punto par e impar;
- 30 la Figura 13 muestra la orientación de formatos de impresión en base a la imagen CFA;
la Figura 14 muestra un diagrama de bloques de la cadena de captura de imagen;
la Figura 15 muestra la disposición de píxeles en un mosaico Bayer CFA 2G;
- 35 la Figura 16 muestra el proceso RGB de linealización;
la Figura 17 muestra el proceso RGB de planarización;
- 40 la Figura 18 muestra un diagrama de bloques de la cadena de impresión de imagen;
la Figura 19 muestra un rango de color de muestra para un plano de color simple;
la Figura 20 muestra las etapas involucradas en el balance del blanco y en la expansión de rango;
- 45 la Figura 21 muestra un diagrama de bloques de un aparato capacitado para realizar el balance del blanco y la expansión de rango;
la Figura 22 muestra los diversos píxeles de plano de color en relación con una resolución CFA;
- 50 la Figura 23 muestra el efecto de girar el plano del verde en 45 grados;
la Figura 24 muestra la distancia entre los píxeles girados para el plano del verde;
- 55 la Figura 25 muestra el proceso de movimiento de mapeo en el espacio CFA no girado respecto al espacio CFA girado;
la Figura 26 muestra un diagrama de bloques del proceso de intensificación;
- 60 la Figura 27 muestra el proceso involucrado en el filtrado pasa-alto de un pixel de luminancia simple con una función de 3 x 3;
la Figura 28 muestra la transformación de conversión de RGB a CMY;
- 65 la Figura 29 muestra la conversión de RGB a CMY mediante interpolación tri-lineal;
la Figura 30 muestra la replicación de pixel, de un pixel simple a un bloque de 5 x 5;

ES 2 282 136 T3

la Figura 31 muestra un diagrama de bloques del proceso de generación de semitono;

la Figura 32 muestra el proceso de puntos de reformato para la impresora;

5 la Figura 33 muestra un diagrama de bloques de la unidad de captura de imágenes;

la Figura 35 muestra un diagrama de bloques de la unidad de acceso de imagen;

10 la Figura 36 muestra un diagrama de bloques de la unidad de histograma de imagen;

la Figura 37 muestra un diagrama de bloques de la interfaz de impresora;

la Figura 38 muestra el diagrama de bloques de la interfaz Memjet;

15 la Figura 39 muestra la generación de anchos de pulso AEnable y BEnable;

la Figura 40 muestra un diagrama de bloques de la lógica de conteo de punto;

20 la Figura 41 muestra la interfaz de la unidad de generador de impresión;

la Figura 42 muestra un diagrama de bloques de la unidad de generador de impresión;

la Figura 43 muestra un diagrama de bloques de la unidad de acceso de patrón de prueba;

25 la Figura 44 muestra un diagrama de bloques de la Memoria Intermedia 5;

la Figura 45 muestra un diagrama de bloques de la Memoria Intermedia 4;

30 la Figura 46 muestra un diagrama de bloques del proceso de UpInterpolate, Halftone y Reformat;

la Figura 47 muestra cómo realizar el mapeo desde una célula vibratoria hasta una célula vibratoria acumulada;

la Figura 48 muestra un diagrama de bloques del proceso de Convertir RGB a CMY;

35 la Figura 49 muestra un diagrama de bloques de la Memoria Intermedia 2;

la Figura 50 muestra un filtro espacial básico pasa-alto que utiliza una función 3 x 3;

40 la Figura 51 muestra un diagrama de bloques de la unidad de intensificación;

la Figura 52 muestra la estructura de la memoria Intermedia 1;

la Figura 53 muestra un diagrama de bloques del proceso de Remuestrear y Crear Canal de Luminancia;

45 la Figura 54 muestra un diagrama de bloques de la Unidad de Convolución;

la Figura 55 muestra el orden de píxeles generados desde el receptor;

50 la Figura 56 muestra el movimiento en "x" o en "y" en el espacio girado y no girado;

la Figura 57 muestra la dirección de entradas en la sub-memoria del verde de la Memoria Intermedia 1;

la Figura 58 muestra la relación entre las entradas del verde en función de la rotación;

55 la Figura 59 muestra un muestreo 4 x 4 del canal del verde;

la Figura 60 muestra un muestreo 4 x 4, tipo 1, del verde;

60 la Figura 61 muestra un muestreo 4 x 4, tipo 2, del verde;

la Figura 62 muestra los dos tipos de direccionamiento de fila para el verde;

la Figura 63 muestra el direccionamiento de entradas en las sub-memorias del rojo y del azul de la memoria intermedia 1;

65 la Figura 64 muestra las 16 primeras muestras leídas para calcular el primer pixel;

ES 2 282 136 T3

la Figura 65 muestra el peor caso de superposición de lectura 4 x 4 desde las memorias intermedias del azul y del rojo;

la Figura 66 muestra un diagrama de bloques de la unidad de giro, balance del blanco y expansión de rango, y

la Figura 67 muestra el área de imagen activa del interior del espacio coordinado generado.

1. Compendio del PCP

1.1 Compendio Funcional de Alto Nivel

El Procesador Central de la Printcam (PCP) posee toda la potencia de procesamiento para una Printcam, y se ha diseñado específicamente para ser usado en el sistema de cámara digital Printcam. El PCP 3 conecta un sensor 1 de imagen (para la captura de imágenes), y una impresora 2 Memjet para la impresión de imágenes. En términos de procesamiento de imagen, se puede pensar que el PCP es el traductor de imágenes desde la captura hasta la impresión, como se muestra en la Figura 1:

- El Sensor de Imagen 1 es un sensor de imagen CMOS, que captura una imagen RGB de 1500 x 1000. El Sensor de Imagen constituye el dispositivo de entrada.
- La Cabeza de Impresión 2 es una impresora Memjet de 1600 dpi de 10,16 cm (4 pulgadas) de largo, capacitada para imprimir en tres colores: cyan, magenta y amarillo. La Cabeza de Impresión constituye el dispositivo de salida de imagen.
- El PCP 3 toma una imagen desde el Sensor de Imagen 1, la procesa, y envía la forma final de la imagen a la Cabeza de Impresión 2, para su impresión. Puesto que el Sensor de Imagen 1 captura en RGB, y la Cabeza de Impresión 2 imprime en CMY, el PCP 3 debe traducir desde el espacio de color RGB hasta el espacio de color CMY. El PCP 3 contiene todos los requisitos para el procesamiento intermedio de la imagen, incluyendo el balance del blanco, la corrección de color y el mapeo de gama de frecuencia, la intensificación de imagen, y la generación de semitono. Adicionalmente, el PCP 3 controla la interfaz de usuario y el proceso de impresión completo, proporcionando soporte para una diversidad de formatos de imagen. El PCP 3 contiene también interfaces para permitir exportar e importar fotos, cumpliendo con el estándar DPOF (Digital Print Order Format, Formato Digital de Órdenes de Impresión).

1.2 Compendio Interno de Alto Nivel

El PCP 3 está diseñado para ser fabricado utilizando un proceso CMOS de 0,25 micrómetros, con aproximadamente 10 millones de transistores, casi la mitad de los cuales son memoria flash o RAM estática. Esto conduce a un área estimada de 16 mm². El coste de fabricación estimado es de 4 \$ en el año 2001. El PCP 3 es un diseño relativamente directo, y el esfuerzo de diseño puede reducirse con el uso de técnicas de compilación de rutas de datos, macro-células y núcleos IP. El PCP 3 contiene:

- Una CPU/ núcleo microcontrolador 10 de baja velocidad
- 1,5 MBytes de Memoria flash 11 multi-nivel (2 bits por celda)
- una Interfaz 98 de Sensor de Imagen CMOS en el interior de una Unidad de Captura de Imagen 12
- 16 KBytes de memoria flash 13 para almacenamiento de programas
- 4 KBytes de RAM 14 para almacenamiento variable de programas.

El PCP 3 ha sido previsto para que corra a una velocidad de reloj de aproximadamente 100 MHz a 3 V externamente y 1,5 V internamente para minimizar el consumo de energía. La frecuencia operativa real será un múltiplo entero de la frecuencia operativa de de la Cabeza de Impresión. La CPU 10 está previsto que sea una CPU de tipo microcontrolador simple, que corra a aproximadamente 1 MHz. Tanto la CPU 10 como la interfaz 12 de sensor CMOS, pueden ser núcleos suministrados por un proveedor.

La Figura 2 muestra un diagrama de bloques del PCP 3 por separado.

El PCP 3 está diseñado para su uso en sistemas de Printcam. La Figura 3 muestra un diagrama de bloques del PCP 3 conectado al resto del hardware de la Printcam.

ES 2 282 136 T3

2. Antecedentes de la cabeza de impresión

El PCP 3 se ha diseñado específicamente para conectarse a una cabeza de impresión 2 Memjet de 10 cm (4 pulgadas). La cabeza de impresión 2 se utiliza como impresora de página a lo ancho, produciendo una imagen impresa de 10 cm (4 pulgadas) de anchura sin que tenga que ser movida. Al contrario, el papel 20 se imprime según se mueve éste al pasar por la cabeza de impresión 2, como se muestra en la Figura 4.

2.1 Composición de Cabeza de Impresión de 10 cm (4 pulgadas)

Cada cabeza de impresión 2 de 10 cm (4 pulgadas) consiste en 8 segmentos, siendo cada segmento de 12,7 mm (1/2 pulgada) de longitud. Cada uno de los segmentos 21 imprime puntos cyan, magenta y amarillo bi-nivel sobre una parte diferente de la página, para producir la imagen final. Las posiciones de los segmentos se muestran en la Figura 5.

Puesto que la cabeza de impresión 2 imprime puntos a 1600 dpi, cada punto tiene un diámetro de 22,5 μm , y están separados por 15,875 μm . Así, cada segmento de 12,7 mm (media pulgada) imprime 800 puntos, correspondiendo los 8 segmentos a las posiciones:

TABLA 1

Puntos de Imagen Final Dirigidos por Cada Segmento

Segmento	Primer punto	Último punto
0	0	799
1	800	1.599
2	1.600	2.399
3	2.400	3.199
4	3.200	3.999
5	4.000	4.799
6	4.800	5.599
7	5.600	6.399

Aunque cada segmento 21 produce 800 puntos de la imagen final, cada punto está representado por una combinación de tinta cyan, magenta y amarilla bi-nivel. Puesto que la impresión es bi-nivel, la imagen de entrada será vibrada o de error difuso para unos mejores resultados.

Cada segmento 21 contiene entonces 2.400 boquillas: 800 de cada color cyan, magenta y amarillo. Una cabeza de impresión 2 de 10 cm (cuatro pulgadas) contiene 8 segmentos 21 con un total de 19.200 boquillas.

2.1.1 Agrupamiento de Boquillas Dentro de un Segmento

Las boquillas 22 en el interior de un segmento 21 simple, están agrupadas por motivos de estabilidad física, así como también por reducción del consumo de potencia durante la impresión. En términos de estabilidad física, un total de 10 boquillas comparten el mismo depósito de tinta. En términos de consumo de potencia, los agrupamientos se realizan de manera que permiten un modo de presión de baja velocidad y uno de alta velocidad.

La cabeza de impresión 2 soporta dos velocidades de impresión para permitir que se lleven a cabo diferentes relaciones de velocidad/ potencia en diferentes configuraciones de productos.

En el modo de impresión de baja velocidad, 96 boquillas 22 son disparadas simultáneamente desde cada cabeza de impresión 2 de 10 cm (4 pulgadas). Las boquillas disparadas deben ser maximalmente distantes, de modo que se disparen 12 boquillas 22 desde cada segmento. Para disparar todas las 19.200 boquillas, se deben disparar 200 conjuntos diferentes de 96 boquillas.

En el modo de impresión de alta velocidad, se disparan 192 boquillas 22 simultáneamente desde cada cabeza de impresión 2 de 10 cm (4 pulgadas). Las boquillas 22 disparadas deben ser maximalmente distantes, de modo que se disparen 24 boquillas desde cada segmento. Para disparar la totalidad de 19.200 boquillas, se deben disparar 100 conjuntos diferentes de 192 boquillas.

ES 2 282 136 T3

El consumo de potencia en el modo de baja velocidad es la mitad que en el modo de alta velocidad. Obsérvese, no obstante, que la energía consumida para imprimir una línea, y por tanto una página, es la misma en ambos casos.

5 En un escenario tal como una Printcam alimentada con batería, los requisitos de consumo de potencia determinan el uso de la impresión de baja velocidad.

2.1.1.1 10 Boquillas Forman un Pod

10 Un pod simple 23 consiste en 10 boquillas 22 que comparten un depósito 5 común de tinta. 5 boquillas 22 están en una fila, y 5 están en otra. Cada boquilla 22 produce puntos de aproximadamente $22,5 \mu\text{m}$ de diámetro, separados sobre una parrilla de $15,875 \mu\text{m}$. La Figura 6 muestra la disposición de un pod simple, con las boquillas 22 numeradas de acuerdo con el orden en el que las mismas deben ser disparadas.

15 Aunque las boquillas 22 son disparadas por este orden, la relación de boquillas 22 y la colocación física de puntos sobre la página impresa, es diferente. Las boquillas 22 de una fila representan los puntos pares de una línea de la página, y las boquillas de la otra fila representan los puntos impares de la línea adyacente de la página. La Figura 7 muestra el mismo pod 23 con las boquillas 22 numeradas de acuerdo con el orden en el que las mismas deben ser cargadas.

20 Las boquillas 22 interiores a un pod 23, están por tanto lógicamente separadas por la anchura de 1 punto. La distancia exacta entre las boquillas 22 dependerá de las propiedades del mecanismo de disparo Memjet. La cabeza de impresión 2 está diseñada con boquillas escalonadas diseñadas para emparejarse con el flujo del papel 20.

2.1.1.2 3 Pods Forman un Cromo-Pod

25 Un pod 23 de cada color (cyan, magenta y amarillo), se agrupan en un cromapod 24. Un cromapod 24 representa componentes de diferente color del mismo conjunto horizontal de 10 puntos, sobre líneas diferentes. La distancia exacta entre diferentes pods 23 de color depende de los parámetros operativos de la Memjet, y puede variar de un diseño de Memjet a otro. La distancia se considera que es una cantidad constante de anchuras de punto, y debe ser
30 tomada en cuenta por lo tanto cuando se realiza la impresión: los puntos imprimidos por las boquillas cyan serán para líneas diferentes que los imprimidos por las boquillas magenta o amarillo. El algoritmo de impresión debe permitir una distancia variable de hasta aproximadamente 8 anchuras de punto entre colores (véase la Tabla 3 para más detalles). La Figura 8 ilustra un cromapod 24 simple.

35 2.1.1.3 5 Cromapods Forman un Grupo-Pod

5 cromapods 24 se organizan en un solo grupopod 25. Puesto que cada cromapod contiene 30 boquillas 22, cada grupopod contiene 150 boquillas 22: 50 boquillas cyan, 50 boquillas magenta y 50 boquillas amarillo. La disposición se muestra en la Figura 9, con los cromapods numerados como 0-4. Obsérvese que la distancia entre cromapods
40 adyacentes se ha exagerado por motivos de claridad.

2.1.1.4 2 Grupo-Pods Forman un Grupo-Fase

45 2 grupopods 25 se organizan en un único grupofase 26. El grupofase 26 se denomina de ese modo debido a que los grupos de boquillas 23 del interior de un grupofase se disparan simultáneamente durante una fase de disparo dada (esto se explica con mayor detalle en lo que sigue). La formación de un grupofase a partir de 2 grupopods 25, se realiza totalmente a efectos de impresión a baja velocidad y a alta velocidad a través de 2 líneas PodgroupEnable.

50 Durante la impresión a baja velocidad, se dispone solamente una de las dos líneas PodgroupEnable en un pulso de disparo dado, de modo que solamente uno grupopod de los dos dispara las boquillas. Durante la impresión a alta velocidad, se disponen ambas líneas PodgroupEnable, de modo que ambos grupopods disparan las boquillas. En consecuencia, la impresión a baja velocidad es dos veces más larga que la impresión a alta velocidad, puesto que la impresión a alta velocidad dispara el doble de boquillas a la vez.

55 La Figura 10 ilustra la composición del grupofase. La distancia entre grupopods adyacentes ha sido exagerada por motivos de claridad.

2.1.1.5 2 Grupo-Fases forman un Grupo-de-Disparo

60 Dos grupofases (PhasegroupA y PhasegroupB) se organizan en un sólo grupo de disparo 27, con 4 grupos de disparo en cada segmento. Los grupos de disparo 27 se denominan así debido a que todos ellos disparan las mismas boquillas 27 simultáneamente. Dos líneas de habilitación, AEnable y BEnable, permiten el disparo de las boquillas PhasegroupA y de las boquillas PhasegroupB independientemente como fases de disparo diferentes. La disposición se muestra en la Figura 11. La distancia entre agrupamientos adyacentes se ha exagerado por motivos de
65 claridad.

ES 2 282 136 T3

2.1.1.6 Sumario de Agrupamiento de Boquillas

La Tabla 2 es un sumario de los agrupamientos de boquillas en una cabeza de impresión.

5

TABLA 2

Agrupamientos de Boquillas para una cabeza de Impresión simple de 10 cm (4 pulgadas)

10

Nombre del Agrupamiento	Composición	Relación de Replicación	Conteo de Boquilla
Boquilla 22	Unidad de base	1:1	1
Pod 23	Boquillas por pod	10:1	10
Croma-pod 24	Pods por croma-pod CMY	3:1	30
Grupo-pod 25	Croma-pods por grupo-pod	5:1	150
Grupo-fase 26	Grupo-pods por grupo-fase	2:1	300
Grupo de disparo 27	Grupo-fases por grupo de disparo	2:1	600
Segmento 21	Grupos de disparo por segmento	4:1	2.400
Cabeza de Impresión 2 de 10 cm (4 pulgadas)	Segmentos por cabeza de impresión de 10 cm (4 pulgadas)	8:1	19.200

25

2.2 Ciclos de Carga e Impresión

30

Una cabeza de impresión 2 simple de 10 cm (4 pulgadas) contiene un total de 19.200 boquillas 22. Un Ciclo de Impresión incluye el disparo de hasta la totalidad de estas boquillas, dependiendo de la información que se ha de imprimir. Un Ciclo de Carga incluye la carga de la cabeza de impresión con la información que ha de ser imprimida durante el posterior Ciclo de Impresión.

35

Cada boquilla 22 posee un bit de NozzleEnable asociado, que determina si la boquilla se ha de disparar o no durante el Ciclo de Impresión. Los bits de NozzleEnable (uno por boquilla) se cargan por medio de un conjunto de registros de desplazamiento.

40

Lógicamente, existen 3 registros de desplazamiento por segmento (uno por color), cada uno con una longitud de 800. Según se desplazan los bits por el registro de desplazamiento para un color dado, son dirigidos hasta las boquillas inferior y superior sobre pulsos alternos. Internamente, cada registro de desplazamiento con una profundidad de 800, comprende dos registros de desplazamiento de 400 de profundidad: uno para las boquillas superiores y uno para las boquillas inferiores. Los bits alternos son desplazados hacia los registros internos. En lo que se refiere a la interfaz externa, sin embargo, existe un único registro de desplazamiento de 800 de profundidad.

45

Una vez que todos los registros de desplazamiento han sido cargados completamente (800 pulsos de carga), todos los bits son transferidos en paralelo a los bits de NozzleEnable apropiados. Esto equivale a una transferencia paralelo simple de 19.200 bits. Una vez que la transferencia ha tenido lugar, el Ciclo de Impresión puede empezar. El Ciclo de Impresión y el Ciclo de Carga pueden ocurrir simultáneamente mientras se produce la carga en paralelo de todos los bits de NozzleEnable al final del Ciclo de Impresión

50

2.2.1 Ciclo de Carga

55

El Ciclo de Carga está relacionado con la carga de los registros de desplazamiento de la cabeza de impresión con los siguientes bits de NozzleEnable del Ciclo de Impresión.

60

Cada segmento 21 posee 3 entradas relacionadas directamente con los registros de desplazamiento del cian, el magenta y el amarillo. Estas entradas se denominan CDatain, MDatain e YDatain. Puesto que existen 8 segmentos, existe un total de 24 líneas de entrada de color por cabeza de impresión de 10 cm (4 pulgadas). Un pulso simple por la línea SRClock (compartido entre la totalidad de los 8 segmentos), transfiere los 24 bits a los registros de desplazamiento apropiados. Pulsos alternos transfieren bits a las boquillas inferior y superior, respectivamente. Puesto que existen 19.200 boquillas, se requiere un total de 800 pulsos para la transferencia. Una vez que se ha transferido la totalidad de los 19.200 pulsos, un pulso simple por la línea PTransfer compartida, provoca la transferencia de datos paralelo desde los registros de desplazamiento a los bits NozzleEnable apropiados.

65

ES 2 282 136 T3

La transferencia paralelo a través de un pulso sobre PTransfer, debe tener lugar después de que haya acabado el Ciclo de Impresión. En otro caso, los bits de NozzleEnable para la línea que se está imprimiendo serán incorrectos.

Puesto que se cargan los 8 segmentos 21 con un simple pulso SRClock, cualquier procedimiento de impresión debe producir los datos en la secuencia correcta para la cabeza de impresión. Como ejemplo, el primer pulso SRClock transferirá los bits CMY para el siguiente punto 0, 800, 1600, 2400, 3200, 4000, 4800 y 5600 del Ciclo de Impresión. El segundo pulso SRClock transferirá los bits CMY para el siguiente punto 1, 801, 1601, 2401, 3201, 4001, 4801 y 5601 del Ciclo de Impresión. Tras los 800 pulsos SRClock, se puede proporcionar el pulso PTransfer.

Es importante observar que las salidas CMY impar y par, aunque imprimidas durante el mismo Ciclo de Impresión, no aparecen por la misma línea física de salida. La separación física de boquillas impar y par dentro de la cabeza de impresión, así como la separación entre boquillas de diferentes colores, asegura que producirán puntos sobre líneas diferentes de la página. La diferencia relativa debe ser tenida en cuenta cuando se cargan los datos en la cabeza de impresión. La diferencia real en líneas depende de las características del mecanismo de chorro de tinta que se utilice en la cabeza de impresión. Las diferencias pueden estar definidas por variables D_1 y D_2 , donde D_1 es la distancia entre boquillas de diferentes colores, y D_2 es la distancia entre boquillas del mismo color. La Tabla 3 muestra los puntos transferidos al segmento n de una cabeza de impresión sobre los primeros 4 pulsos.

TABLA 3

Orden de Puntos Transferidos a una Cabeza de Impresión de 10 cm (4 pulgadas)

Pulso	Punto	Línea Amarillo	Línea Magenta	Línea Cyan
1	$800S^a$	N	$N + D_1^b$	$N + 2D_1$
2	$800S + 1$	$N + D_2^c$	$N + D_1 + D_2$	$N + 2D_1 + D_2$
3	$800S + 2$	N	$N + 2D_1$	$N + 2D_1$
4	$800S + 3$	$N + D_2$	$N + D_1 + D_2$	$N + 2D_1 + D_2$

a. S = número de segmento (0-7)

b. D_1 = número de líneas entre las boquillas de un color y el siguiente (probablemente = 4-8)

c. D_2 = número de líneas entre dos filas de boquillas del mismo color (probablemente = 1)

Y así sucesivamente para la totalidad de los 800 pulsos.

Los datos pueden ser registrados en la cabeza de impresión a una velocidad máxima de 20 MHz, con lo que se cargará la totalidad de los datos para la siguiente línea en 40 μ s.

2.2.2 Ciclo de Impresión

Una cabeza de impresión 2 de 10 cm (4 pulgadas) contiene 19.200 boquillas 22. Dispararlas todas a la vez podría consumir demasiada potencia y ser problemático en términos de relleno de tinta y de interferencia de boquilla. En consecuencia, se definen dos modos de disparo: un modo de impresión de baja velocidad y un modo de impresión de alta velocidad:

- En el modo de impresión de baja velocidad, existen 200 fases, disparando cada fase 96 boquillas. Esto equivale a 12 boquillas por segmento, o 3 por grupo de disparo.
- En el modo de impresión de alta velocidad, existen 100 fases, disparando cada fase 192 boquillas. Esto equivale a 24 boquillas por segmento, o 6 grupos de disparo.

Las boquillas que van a ser disparadas durante un pulso dado, se determinan mediante:

- 3 bits ChromapodSelect (seleccionan 1 de 5 cromapods 24 a partir de un grupo de disparo 27)
- 4 bits NozzleSelect (seleccionan 1 de 10 boquillas 22 a partir de un pod 23)
- 2 bits de líneas PodgroupEnable (seleccionan 0, 1 ó 2 grupo-pods 25 para el disparo).

Cuando se establece una de las líneas PodgroupEnable, solamente las 4 boquillas especificadas del Podgroup se dispararán según se determina mediante ChromapodSelect y NozzleSelect. Cuando se determinan ambas líneas PodgroupEnable, ambos grupos pod dispararán sus boquillas. Para el modo de baja velocidad, se requieren dos pulsos de disparo, con PodgroupEnable = 10 y 01, respectivamente. Para el modo de alta velocidad, solamente se requiere un pulso de disparo, con PodgroupEnable = 11.

ES 2 282 136 T3

La duración del pulso de disparo viene dada por las líneas AEnable y BEnable, las cuales disparan las boquillas PhasegroupA y PhasegroupB de todos los grupos de disparo, respectivamente. La duración típica de un pulso de disparo es de 1,3 - 1,8 μ s respectivamente. La duración de un pulso depende de la viscosidad de la tinta (dependiente de la temperatura y de las características de la tinta), y de la cantidad de potencia disponible respecto a la cabeza de impresión. Véase la Sección 2.3 para los detalles sobre los informes desde la cabeza de impresión a efectos de compensación del cambio de temperatura.

Las líneas AEnable y BEnable son líneas separadas con el fin de que los pulsos de disparo puedan solaparse. De ese modo, las 200 fases de un Ciclo de Impresión de baja velocidad consisten en 100 fases A y 100 fases B, que proporcionan de forma efectiva 100 conjuntos de Fase A y de Fase B. De igual modo, las 100 fases de un ciclo de impresión de alta velocidad consisten en 50 fases A y 50 fases B, que proporcionan de manera efectiva 50 fases de la fase A y de la fase B.

La Figura 12 muestra las líneas AEnable y BEnable durante un Ciclo de Impresión típico. En una impresión de alta velocidad, existen 50 ciclos de 2 μ s, mientras que en una impresión de baja velocidad existen 100 ciclos de 2 μ s.

Para el modo de impresión de alta velocidad, el orden de disparo es:

- ChromapodSelect 0, NozzleSelect 0, PodgroupEnable 11 (Fases A y B)
- ChromapodSelect 1, NozzleSelect 0, PodgroupEnable 11 (Fases A y B)
- ChromapodSelect 2, NozzleSelect 0, PodgroupEnable 11 (Fases A y B)
- ChromapodSelect 3, NozzleSelect 0, PodgroupEnable 11 (Fases A y B)
- ChromapodSelect 4, NozzleSelect 0, PodgroupEnable 11 (Fases A y B)
- ChromapodSelect 0, NozzleSelect 1, PodgroupEnable 11 (Fases A y B)
-
- ChromapodSelect 3, NozzleSelect 9, PodgroupEnable 11 (Fases A y B)
- ChromapodSelect 4, NozzleSelect 9, PodgroupEnable 11 (Fases A y B)

Para el modo de impresión de baja velocidad, la orden de disparo es similar. Para cada fase del modo de alta velocidad, en la que PodgroupEnable era 11, se sustituyen dos fases de PodgroupEnable = 01 y 10, como sigue:

- ChromapodSelect 0, NozzleSelect 0, PodgroupEnable 01 (Fases A y B)
- ChromapodSelect 0, NozzleSelect 0, PodgroupEnable 10 (Fases A y B)
- ChromapodSelect 1, NozzleSelect 0, PodgroupEnable 01 (Fases A y B)
- ChromapodSelect 1, NozzleSelect 0, PodgroupEnable 10 (Fases A y B)
-
- ChromapodSelect 3, NozzleSelect 9, PodgroupEnable 01 (Fases A y B)
- ChromapodSelect 3, NozzleSelect 9, PodgroupEnable 10 (Fases A y B)
- ChromapodSelect 4, NozzleSelect 9, PodgroupEnable 01 (Fases A y B)
- ChromapodSelect 4, NozzleSelect 9, PodgroupEnable 10 (Fases A y B)

Cuando se dispara una boquilla 22, tarda aproximadamente 100 μ s en rellenarse. La boquilla 22 no puede ser disparada antes de que haya transcurrido este tiempo de rellenado. Esto limita la velocidad de impresión más rápida a 100 μ s por línea. En el modo de impresión de alta velocidad, el tiempo para imprimir una línea es de 100 μ s, de modo que el tiempo entre el disparo de una boquilla de una línea a la siguiente se equipara al tiempo de rellenado, lo que hace que el modo de impresión de alta velocidad sea aceptable. El modo de impresión de baja velocidad es más lento que éste, de modo que también es aceptable.

El disparo de una boquilla 22 causa también perturbaciones acústicas durante un tiempo limitado en el interior de un depósito común de tinta del pod 23 de la boquilla. Las perturbaciones pueden interferir con el disparo de otra boquilla dentro del mismo pod 23. En consecuencia, el disparo de boquillas dentro de un pod deberá separarse una de

ES 2 282 136 T3

otra tanto como sea posible. Por lo tanto, se disparan tres boquillas de un cromapod 24 (una boquilla 22 por color) y a continuación se mueven hacia el siguiente cromapod 24 dentro del grupo-pod 25.

- En el modo de impresión de baja velocidad, los grupo-pods se disparan por separado. Así, los 5 cromapods interiores a ambos grupo-pods deben dispararse todos antes de que el primer cromapod se dispare de nuevo, totalizando ciclos de $10 \times 2 \mu\text{s}$. En consecuencia, cada pod 23 se dispara una vez por cada $20 \mu\text{s}$.
- En el modo de impresión de alta velocidad, los grupo-pods 25 se disparan conjuntamente. De ese modo, los 5 cromapods 24 interiores a un grupo-pod simple deben dispararse todos antes de que el primer cromapod se dispare de nuevo, totalizando ciclos de $5 \times 2 \mu\text{s}$. En consecuencia, cada pod 23 se dispara una vez cada $10 \mu\text{s}$.

Como el canal de tinta tiene $300 \mu\text{m}$ de largo y la velocidad del sonido en la tinta es de alrededor de 1500 m/s , la frecuencia de resonancia del canal de tinta es de $2,5 \text{ MHz}$, con lo que el modo de baja velocidad permite 50 ciclos de resonancia para amortiguar el pulso acústico, y el modo de alta velocidad permite 25 ciclos de resonancia. Así, cualquier interferencia acústica es mínima en ambos casos.

2.2.3 Tiempo de Muestra

Como ejemplo, considérese el tiempo de impresión de una foto de $10 \times 15 \text{ cm}$ ($4'' \times 6''$) en 2 segundos, como se requiere por parte de la Printcam. Con el fin de imprimir una foto en 2 segundos, la cabeza de impresión de 10 cm (4 pulgadas) debe imprimir 9.600 líneas (6×1.600). El redondeo a 10.000 líneas en 2 segundos produce un tiempo de línea de $200 \mu\text{s}$. Un Ciclo de Impresión simple y un Ciclo de Carga único deben acabar, ambos, dentro de este tiempo. Adicionalmente, el proceso físico externo a la cabeza de impresión debe mover el papel una cantidad apropiada.

Desde el punto de vista de la impresión, el modo de impresión de baja velocidad permite que una cabeza de impresión de 10 cm (4 pulgadas) imprima una línea completa en $200 \mu\text{s}$. En el modo de impresión de baja velocidad, se disparan 96 boquillas 22 por pulso de disparo, habilitando con ello la impresión de la línea completa dentro del tiempo especificado.

Los 800 pulsos SRClock para la cabeza de impresión 2 (cada pulso de reloj transfiere 24 bits), deben tener también lugar dentro del tiempo de línea de $200 \mu\text{s}$. La longitud de un pulso SRClock no puede exceder de $200 \mu\text{s}/800 = 250 \text{ ns}$, lo que indica que la cabeza de impresión debe estar controlada a 4 MHz . Adicionalmente, el tiempo medio para calcular cada valor de bit (para cada una de las 19.200 boquillas) no debe exceder de $200 \mu\text{s}/19.200 = 10 \text{ ns}$. Esto requiere un generador de pulso que corra a una de las siguientes velocidades:

- 100 MHz generando 1 bit (punto) por ciclo
- 50 MHz generando 2 bits (puntos) por ciclo
- 25 MHz generando 4 bits (puntos) por ciclo.

2.3 Informes desde la Cabeza de Impresión

La cabeza de impresión 2 produce varias líneas de información (acumulada desde los 8 segmentos). Las líneas de información se utilizan para ajustar el tiempo de los pulsos de disparo. Aunque cada segmento 21 produce la misma información, los informes procedentes de todos los segmentos comparten las mismas líneas tri-estado de bus. En consecuencia, solamente un segmento 21 puede proporcionar información en cada momento.

Un pulso por la línea SenseSegSelect compuesto con datos sobre Cyan, habilita las líneas de detección para ese segmento. Las líneas de detección de información procederán del segmento seleccionado hasta el siguiente pulso SenseSegSelect. Las líneas de detección de información son como sigue:

- Tsense informa al controlador de cómo está de caliente la cabeza de impresión. Esto permite al controlador ajustar el tiempo de disparo de los pulsos, puesto que la temperatura afecta a la viscosidad de la tinta.
- Vsense informa al controlador sobre cuánta tensión se encuentra disponible para el actuador. Esto permite al controlador compensar una batería de respuesta plana o una fuente de alta tensión ajustando la anchura de pulso.
- Rsense informa al controlador sobre la resistividad (ohmios por cuadro) del calentador del actuador. Esto permite al controlador ajustar los anchos de pulso para mantener una energía constante con independencia de la resistividad del calentador.
- Wsense informa al controlador de la anchura de la parte crítica del calentador, la cual puede variar hasta un $\pm 5\%$ debido a las variaciones litográficas y de grabado químico. Esto permite al controlador ajustar la anchura de pulso de forma apropiada.

ES 2 282 136 T3

2.4 Ciclos especiales

2.4.1 Ciclo de Pre calentamiento

5 El proceso de impresión tiene una fuerte tendencia a mantenerse a la temperatura de equilibrio. Para asegurar que la primera sección de la fotografía impresa tiene un tamaño de punto uniforme, la temperatura de equilibrio debe alcanzarse con anterioridad a la impresión, a todos los puntos. Esto se consigue por medio de un ciclo de pre calentamiento.

10 El ciclo de Pre calentamiento incluye un Ciclo de Carga para todas las boquillas con 1s (es decir, disponiendo todas las boquillas para disparar), y un número de pulsos cortos de disparo para cada boquilla. La duración del pulso debe ser insuficiente para disparar las gotas, pero suficiente para calentar la tinta. En total, se requieren aproximadamente 200 pulsos para cada boquilla, realizando ciclos mediante la misma secuencia que un Ciclo de Impresión estándar.

15 La información durante el modo de Pre calentamiento, viene proporcionada por T_{sense} , y continúa hasta que se alcanza una temperatura de equilibrio (alrededor de 30°C por encima del ambiente). La duración del modo Pre calentamiento es de alrededor de 50 milisegundos, y depende de la composición de la tinta.

20 El pre calentamiento se realiza antes de cada trabajo de impresión. Éste no afecta al rendimiento de la impresora, puesto que se lleva a cabo mientras se están transfiriendo los datos a la impresora.

2.4.2 Ciclo de Limpieza

25 Con el fin de reducir las posibilidades de que las boquillas resulten obstruidas, se puede realizar un ciclo de limpieza con anterioridad a cada trabajo de impresión. Cada boquilla se dispara un número de veces hacia una esponja absorbente.

30 El ciclo de limpieza incluye un sólo Ciclo de Carga para todas las boquillas con 1s (es decir, disponiendo todas las boquillas para disparar), y un número de pulsos de disparo para cada boquilla. Las boquillas son limpiadas por medio de la misma secuencia de disparo de boquilla como en un Ciclo de Impresión estándar. El número de veces que cada boquilla se dispara, depende de la composición de la tinta y del tiempo que la impresora ha estado en vacío, de modo que al igual que con el pre calentamiento, el ciclo de limpieza no tiene ningún efecto sobre el comportamiento de la impresora.

35

(Tabla pasa a página siguiente)

40

45

50

55

60

65

ES 2 282 136 T3

2.5 Sumario de interfaz de cabeza de impresión

Una cabeza de impresión 2 de 10 cm (4 pulgadas) tiene las siguientes conexiones:

TABLA 4

Conexiones de Cabeza de Impresión de 10 cm (cuatro pulgadas)

Nombre	#Patillas	Descripción
ChromapodSelect	3	Seleccionar qué croma-pod disparar (0-4)
NozzleSelect	4	Seleccionar qué boquilla del pod disparar (0-9)
PodgroupEnable	2	Habilitar los pod-groups para disparar (opción de: 01, 10, 11)
AEnable	1	Disparar pulso para grupo de fase A
BEnable	1	Disparar pulso para grupo de fase B
CDataIn[0-7]	8	Introducir cyan en el registro de desplazamiento de cyan de los segmentos 0-7
MDataIn[0-7]	8	Introducir magenta en el registro de desplazamiento de magenta de los segmentos 0-7
YDataIn[0-7]	8	Introducir amarillo en el registro de desplazamiento de amarillo de los segmentos 0-7
SRClock	1	Un pulso por SRClock (ShiftRegisterClock) carga los valores actuales desde CDataIn[0-7], MDataIn[0-7] e YDataIn[0-7] en los 24 registros de desplazamiento
PTransfer	1	Transferencia paralelo de datos desde los registros de desplazamiento hasta los bits NozzleEnable internos (uno por boquilla)
SenseSegSelect	1	Un pulso por SenseSegSelect Compuesto con datos sobre CDataIn[n] selecciona las líneas de detección para el segmento n.
Tsense	1	Detección de temperatura
Vsense	1	Detección de tensión
Rsense	1	Detección de resistividad
Wsense	1	Detección de anchura
Lógica GND	1	Tierra lógica
Lógica PWR	1	Potencia lógica
V ⁻	Barras de Bus	Tierra del actuador
V ⁺	Barras de Bus	Potencia del actuador
TOTAL	44	

ES 2 282 136 T3

En el interior de la cabeza de impresión de 10 cm (4 pulgadas), cada segmento tiene las conexiones que siguen con los adaptadores de enlace:

TABLA 5

Conexiones de Segmento Internas de Cabeza de Impresión de 10 cm (cuatro pulgadas)

Nombre	#Patillas	Descripción
ChromapodSelect	3	Seleccionar qué cromapod disparar (0-4)
NozzleSelect	4	Seleccionar qué boquilla del pod disparar (0-9)
PodgroupEnable	2	Habilitar los grupo-pods para disparar (opción de: 01, 10, 11)
AEnable	1	Disparar pulso para grupo de fase A
BEnable	1	Disparar pulso para grupo de fase B
CDataIn	1	Introducir cyan en el registro de desplazamiento de cyan
MDataIn	1	Introducir magenta en el registro de desplazamiento de magenta
YDataIn	1	Introducir amarillo en el registro de desplazamiento de amarillo
SRClock	1	Un pulso sobre SRClock (ShiftRegisterClock) carga los valores de corriente desde CDataIn, MDataIn e YDataIn en los 3 registros de desplazamiento
PTransfer	1	Transferencia paralelo de datos desde los registros de desplazamiento hasta los bits de NozzleEnable internos (uno por boquilla)
SenseSegSelect	1	Un pulso sobre SenseSegSelect Compuesto con datos sobre CDataIn selecciona las líneas de detección para este segmento
Tsense	1	Detección de temperatura
Vsense	1	Detección de tensión
Rsense	1	Detección de resistividad
Wsense	1	Detección de anchura
Lógica GND	1	Tierra lógica
Lógica PWR	1	Potencia lógica
V ⁻	21	Tierra del actuador
V ⁺	21	Potencia del actuador
TOTAL	65	(65 x 8 segmentos = 520 para todos los segmentos)

3. Cadenas de procesamiento de imagen

Las secciones anteriores tienen relación solamente con el compendio de la funcionalidad PCP que es de nivel más alto que la del mapeo de imágenes CFA respecto a una diversidad de formatos de salida de impresión. De hecho, existe un número de etapas involucradas en la toma de una imagen desde el sensor de imagen, y en la producción de una impresión de salida de alta calidad. Nosotros hemos fraccionado el proceso de alto nivel en dos cadenas de procesamiento de imagen, cada una de ellas con un número de etapas:

- Cadena de Captura de Imagen
- Cadena de Impresión.

ES 2 282 136 T3

La Cadena de Captura de Imagen se refiere a la captura de la imagen desde el Sensor de Imagen, y a su almacenamiento localmente en el interior de la Printcam. La Cadena de Impresión está relacionada con la toma de la imagen almacenada para la impresión de la misma. Estas dos cadenas mapean sobre la funcionalidad de la Printcam básica como sigue:

- Take&Print = Cadena de Captura de Imagen seguida de la Cadena de Impresión
- Reprint = Cadena de Impresión.

Por ejemplo, un usuario puede imprimir una imagen Thumbnail (Take&Print), y si está contento con los resultados, imprimir varias copias estándar (Reprint).

El capítulo describe una implementación de la cadena de procesamiento de imagen independiente que cumple con los requisitos de calidad de la Printcam. En esta fase, no estamos considerando exactamente cómo se realiza el procesamiento en términos de hardware sino, por el contrario, qué se debe hacer. Estas funciones deben ser mapeadas sobre varias unidades dentro del PCP.

Con independencia de la implementación del PCP, existe un número de limitaciones:

- La imagen de entrada es una imagen RGB contone basada en CFA.
- La imagen de salida es para una cabeza de impresión Memjet (puntos bi-nivel a 1.600 dpi) en un espacio de color CMY, y siempre tiene la misma anchura de salida (10 cm de ancho (4 pulgadas)).

3.0.1 Formatos de Impresión Soportados

El PCP 3 soporta una diversidad de formatos de salida de impresión, como se muestra en la Tabla 6. En todos los casos, la anchura de la imagen es de 10 cm (4 pulgadas) (equiparable a la anchura de la cabeza de impresión). Solamente varía la longitud de la impresión.

TABLA 6

Formatos de Imagen Soportados

Nombre del Formato	Relación de Aspecto	Tamaño de Salida en cm (pulgadas)	Resolución de Salida	Rotación
Standar 30	2:3	10 x 15 (4" x 6")	6.400 x 9.600	90
Passport 31	2:3	10 x 15 (4" x 6")	6.400 x 9.600	90
Panoramic 33	4:6	10 x 30 (4" x 12")	6.400 x 19.200	90
Thumbnail 32	2:3	10x6,8 (4"x2,67")	6.400 x 4.267	0

El sensor de imagen no proporciona información de la orientación. Todas las imágenes de entrada son capturadas a la misma resolución (1.500 x 1.000), y pueden necesitar ser giradas 90 grados con anterioridad a ser imprimidas. La Figura 13 ilustra el mapeo entre la imagen CFA capturada y los diversos formatos de impresión soportados. Obsérvese que aunque la imagen se muestra girada 90 grados en sentido anti-horario, la imagen puede ser girada a favor o en contra de las agujas del reloj.

3.1 Cadena de captura de imagen

La Cadena de Captura de Imagen es responsable de la toma de una imagen desde el Sensor de Imagen, y de almacenarla localmente en el interior de la Printcam. La Cadena de Captura de Imagen incluye un número de procesos que solamente necesitan ser llevados a cabo durante la captura de la imagen. La Cadena de Captura de Imagen ha sido ilustrada en la Figura 14, con secciones consiguientes que detallan los sub-componentes.

3.1.1 Sensor de Imagen 1

La imagen de entrada proviene de un sensor de imagen 1. Aunque se encuentran disponibles una diversidad de sensores, solamente hemos considerado el conjunto de filtro de color (CFA) Bayer. El CFA Bayer posee un número de atributos que aquí se definen.

La imagen capturada por el sensor 1 CMOS (a través de una lente de captación), se supone que ha sido suficientemente filtrada con el fin de eliminar cualquier partícula de solapamiento. El propio sensor dispone de una relación de aspecto de 3:2, con una resolución de 1.500 x 1.000 muestras. La disposición más probable de pixel es el conjunto

ES 2 282 136 T3

de filtro de color (CFA) Bayer, con cada bloque de 2 x 2 píxeles dispuesto en un mosaico 2G según se muestra en la Figura 15.

Cada muestra contone de R, G o B (correspondiente a rojo, verde o azul, respectivamente), es de 10 bits. Obsérvese que cada píxel del mosaico contiene información acerca de solamente uno de los R, G o B. Las estimaciones sobre la falta de información de color deben hacerse antes de que la imagen pueda ser imprimida.

Se considera que el CFA realiza una supresión adecuada de ruido de configuración fija (FPN).

3.1.2 Linealizar RGB 40

- El sensor 40 de imagen es improbable que tenga una respuesta completamente lineal. Por lo tanto, las muestras RGB de 10 bits procedentes del CFA deben ser consideradas como no lineales. Estas muestras no lineales son traducidas a muestras lineales de 8 bits por medio de tablas de búsqueda (una tabla por color).

Los píxeles procedentes de las líneas CFA 0, 2, 4 etc, se indexan en las tablas de R y G, mientras que los píxeles procedentes de las líneas CFA 1, 3, 5, etc, se indexan en las tablas de G y B. Esto es completamente independiente de la orientación de la cámara. El proceso se muestra en la Figura 16. La cantidad total de memoria requerida para cada tabla de búsqueda es de $2^{10} \times 8$ bits. Las 3 tablas de búsqueda 45 requieren por lo tanto un total de 3 Kbytes (3×2^{10} bytes).

3.1.3 Planarizar RGB 41

Los píxeles obtenidos desde el CFA tienen sus planos de color intercalados debido a la naturaleza del mosaico Bayer de píxeles. Con esto queremos decir que sobre líneas horizontales uniformes, un píxel rojo va seguido de un píxel verde y a continuación por otro píxel rojo (los diferentes planos de color están intercalados cada uno con los otros). En algunos sistemas de procesamiento de imágenes, un formato intercalado resulta altamente útil. Sin embargo, en el sistema de procesamiento Printcam, los algoritmos son más eficaces si se trabaja sobre RGB planar.

Una imagen planarizada consiste en una que ha sido separada en sus colores integrantes. En el caso de la imagen RGB de CFA, existen 3 imágenes separadas: una imagen que contiene solamente los píxeles rojos, una imagen que contiene solamente los píxeles azules, y una imagen que contiene solamente los píxeles verdes. Obsérvese que cada plano representa solamente los píxeles de ese color que fueron realmente muestreados. No se realiza ningún remuestreo durante el proceso de planarización. Como resultado, los planos R, G y B son planos no registrados cada uno con los otros, y el plano G es dos veces más grande que cualquiera de los planos R o B. El proceso se muestra en la Figura 17.

El proceso real es muy simple; dependiendo del color de los píxeles leídos, los píxeles de salida son enviados a la siguiente posición en la imagen del plano de color apropiado (por lo tanto, con la misma orientación que el CFA).

Las imágenes planares roja 45 y azul 47, son exactamente la cuarta parte del tamaño de la imagen CFA original. Éstas son exactamente de la mitad de resolución en cada dimensión. Las imágenes roja y azul son por lo tanto de 750×500 píxeles cada una, con una imagen roja implícitamente desviada de la imagen azul por un píxel en el espacio CFA (1.500×1.000) en ambas dimensiones x e y.

Aunque la imagen 45 planar verde es de la mitad de tamaño que la imagen CFA original, no se establece directamente como los planos rojo o azul. La razón se debe a la extensión en forma de tablero de ajedrez del verde. En una línea, el verde es cada píxel impar, y en la siguiente línea, el verde es cada píxel par. Estas líneas alternas del plano verde representan píxeles impares y pares dentro de la imagen CFA. De ese modo, la imagen planar verde es de 750×1.000 píxeles. Ésta tiene ramificaciones para el proceso de remuestreo (véase "Remuestreo en 64" más adelante).

3.1.4 Imagen Almacenada 42

Cada plano de color de la imagen RGB linealizada se escribe en la memoria para su almacenamiento temporal. La memoria deberá ser Flash 11 de modo que la imagen se mantiene después de que se ha interrumpido la alimentación.

La cantidad total de memoria requerida para la imagen RGB lineal planarizada, es de 1.500.000 bytes (aproximadamente 1,5 MB), organizada como sigue:

- R: $750 \times 500 = 375.000$ bytes
- B: $750 \times 500 = 375.000$ bytes
- G: $750 \times 1.000 = 750.000$ bytes.

3.2 Cadena de Impresión

La Cadena de Impresión se refiere a tomar una imagen ya existente desde la memoria 42, e imprimirla en una impresora 2 Memjet. Una imagen se imprime típicamente tan pronto como ha sido capturada, aunque puede ser reimprimida (es decir, sin recaptura).

Existe un número de etapas que son requeridas en la cadena de procesamiento de imagen con el fin de producir impresiones de alta calidad a partir de imágenes capturadas CFA. La Figura 18 ilustra la Cadena de Impresión. La cadena se divide en 3 soluciones de trabajo. La primera es el espacio 50 de captura de imagen original (el mismo espacio que el CFA), la segunda es la resolución 51 intermedia (líneas de 1.280 píxeles de tono continuo), y la resolución final es la resolución 52 de impresora, con líneas de 6.400 puntos bi-nivel.

3.2.1 Imagen de Entrada

La imagen de entrada es una imagen RGB 42 linealizada almacenada en forma planar, según se almacena mediante la Cadena de Captura que se ha descrito en la Sección 3.1.4.

3.2.2 Estadísticas de Acumulación 60

Un número de estadísticas relacionadas con la imagen completa necesitan ser acumuladas con anterioridad a que se puedan realizar procesos tales como el balance del blanco y la expansión de rango. Estas estadísticas necesitan solamente ser acumuladas una vez para todas las impresiones de una imagen 42 capturada particular, y pueden ser acumuladas por separado las imágenes planares del rojo, del verde y del azul.

3.2.2.1 Construcción de Histograma

La primera etapa consiste en construir un histograma para cada valor de 8 bits del plano de color. Cada imagen CFA de 1.500 x 1.000 contiene un total de:

- 375.000 píxeles rojo (contador mínimo requerido de 19 bits)
- 375.000 píxeles azul (contador mínimo requerido de 19 bits)
- 750.000 píxeles verde (contador mínimo requerido de 20 bits).

Por lo tanto, se requiere una simple tabla de 256 x 20 bits para mantener el histograma.

El procedimiento de construcción del histograma es directo, según se ilustra mediante el siguiente pseudo-código:

```

40


---


Para l = 0 a 255
    Entry[l] = 0
45
EndFor
Para Píxel = ImageStart hasta ImageEnd
    P = Image[Píxel]
50
    Entry[p] = Entry[p] + 1
EndFor


---


55

```

3.2.2.2 Determinar Umbrales Alto y Bajo

Una vez que el histograma ha sido construido para el plano de color, puede ser utilizado para determinar un umbral alto y uno bajo. Estos umbrales pueden ser utilizados para automatizar después el balance del blanco y la expansión de rango durante el proceso de impresión.

Basando los umbrales en el número de píxeles del histograma, consideramos que el n% de píxeles más oscuros son expansibles y por lo tanto iguales. De la misma manera, consideramos que el n% de píxeles más brillantes son expansibles y por lo tanto iguales. Se espera que el valor exacto de "n" sea de alrededor del 5%, pero dependerá de las características de respuesta de CFA.

ES 2 282 136 T3

El proceso de determinación del n% de valores más oscuros es directo. Éste incluye un escalonamiento a través del histograma del plano de color a partir del conteo para 0 ascendente (es decir, 0, 1, 2, 3, etc) hasta que se alcance el n% total o nos hayamos desplazado más de una cantidad establecida desde 0. El más alto de estos valores se considera que es el umbral bajo del plano de color. Aunque exista alguna diferencia entre estos valores más oscuros, la diferencia puede ser considerada desechable a los efectos de expansión de rango y balanceo de color.

El proceso de determinación del n% de valores más brillantes, es similar. Éste incluye un escalonamiento a través del histograma del plano de color a partir del conteo para 255 descendente (es decir, 255, 254, 253, etc) hasta que se alcance el n% total o hasta que nos hayamos desplazado más de una cantidad establecida desde 255. El más bajo de estos valores se considera el umbral alto del plano de color. Aunque exista alguna diferencia entre estos valores más brillantes, la diferencia puede ser descartada a los efectos de expansión de rango y de balanceo de color.

La razón para la detención después de una distancia establecida desde 0 o desde 255, consiste en compensar dos tipos de imágenes:

- donde el rango dinámico original sea bajo, o
- donde no exista blanco o negro en la imagen.

En estos dos casos, no deseamos considerar que la totalidad del n% de valores superiores e inferiores sea desechable puesto que tenemos un rango bajo por el que empezar. Podemos establecer de manera segura que los umbrales alto 73 y bajo 72 están fuera del rango de valores de píxeles realmente muestreados. La distancia exacta dependerá del CFA, pero consistirá en dos constantes.

Un rango de color de muestra para un plano de color ha sido mostrado en la Figura 19. Obsérvese que aunque es posible el rango completo de 0-255 para píxeles del plano de color de imagen, esta imagen particular tiene un rango más pequeño. Obsérvese también que el mismo n% de rango 70, 71 de histograma está representado por un rango más grande en el extremo 70 bajo que en el extremo 71 alto. Esto se debe a que el histograma debe contener más píxeles con valores altos más cercanos entre sí que si se compara con el extremo bajo.

Los umbrales alto 73 y bajo 72 deben ser determinados para cada plano de color individualmente. Esta información podrá ser usada para calcular la escala de rango y los factores de desviación que han de ser usados en el proceso posterior de balance del blanco y de expansión de rango.

El siguiente pseudo-código ilustra el proceso de determinación de cualquiera de los dos umbrales (para encontrar el umbral bajo, StartPosition = 255, y Delta = 1. Para encontrar el umbral alto, StartPosition = 0 y Delta = -1). El pseudo-código supone que Umbral es un valor de 8 bits que se reinicia cíclicamente durante la adición

```
Umbral = StartPosition
Total = 0
TotalDelta = 0
Mientras ((TotalDelta < MaxDelta) Y (Total < MaxPixels))
    Umbral = Umbral + Delta
    Total = Total + Entrada[Umbral]
    TotalDelta = TotalDelta + 1
EndWhile
Volver a Umbral
```

3.2.3 Girar Imagen 61

La rotación de la imagen 61 es una etapa opcional en ambos procesos de Captura e Impresión y de Reimpresión.

Diferentes formatos de impresión requieren que la imagen sea girada ya sea 0 o ya sea 90 grados con relación a la orientación CAF, según se muestra en la Figura 13. La cantidad de rotación depende del formato de impresión actualmente seleccionado. Aunque la dirección de rotación no es importante (puede ser a favor de las agujas del reloj o en contra de las agujas del reloj puesto que la nueva orientación solamente se destina a facilitar la anchura de la cabeza de impresión), la dirección de rotación afectará al registro relativo de los 3 planos de color. La Tabla 7 resume la rotación requerida para cada formato de impresión a partir de la orientación CFA original.

TABLA 7

Rotaciones desde la orientación CFA para Formatos de Impresión

Formato de Impresión	Rotación
Estándar 30	90
Passport 31	90
Panoramic 33	90
Thumbnail 32	0

Puesto que estamos girando solamente 90 grados, no se pierde ninguna información durante el proceso de rotación. Para una rotación de 0, la imagen puede ser leída por filas, y para una rotación de 90, la imagen puede ser leída columna por columna. El registro de 3 planos de color debe tener en cuenta la dirección de rotación.

3.2.4 Balance del Blanco y Expansión de Rango 63

Una fotografía se toma pocas veces en condiciones ideales de iluminación. Incluso la noción de “condiciones de iluminación perfectas” está cargada de subjetividad, tanto en términos del fotógrafo como del objeto. Sin embargo, en todos los casos, el objeto de la fotografía se ilumina con luz, ya sea desde una fuente de luz (tal como el sol o una iluminación interior), o ya sea con su propia luz (tal como un rótulo de neón).

En la mayor parte de las condiciones de iluminación, puede parecer al fotógrafo como que es luz “blanca”, lo que habitualmente está lejos de ser luz blanca. La iluminación interior, por ejemplo, típicamente tiene un tono amarillo, y este tono amarillo dará lugar a una fotografía incorrecta. Para la mayor parte de la gente, el tono amarillo sobre la fotografía incorrecta final es erróneo. Aunque puede complementar las condiciones de visión en el momento en que se tomó la fotografía, no iguala el color percibido del objeto. Resulta por tanto crucial realizar un balance del blanco sobre una fotografía con anterioridad a imprimirla.

De la misma manera, una imagen puede ser percibida como de una calidad más alta cuando el rango dinámico de los colores se expande para equipararse con el rango completo en cada plano de color. Esto es particularmente útil de realizar antes de que una imagen sea remuestreada para una resolución más alta. Si el rango dinámico es más alto, se pueden utilizar los valores intermedios en las posiciones interpoladas de los píxeles, evitando una imagen escalonada o fraccionada. La expansión de rango está diseñada para proporcionar el rango de valor 256 completo a aquellos valores realmente muestreados. En el mejor caso, el valor más bajo se mapea a 0, y el valor más alto se mapea a 255. Todos los valores intermedios son mapeados a valores proporcionalmente intermedios entre 0 y 255.

Matemáticamente, la operación realizada consiste en una traslación de Umbral Bajo 72 hasta 0, seguido de un escalado. La fórmula se muestra a continuación:

$$\text{Píxel} = (\text{Píxel} - \text{LowThreshold}) \times \text{RangeScaleFactor}$$

donde:

$$\text{RangeScaleFactor} = \frac{256}{(\text{HighThreshold} - \text{LowThreshold})}$$

RangeScaleFactor deberá estar limitado a un valor máximo para reducir el riesgo de expansión del rango demasiado lejos. Para los detalles sobre el cálculo de LowThreshold 72, véase la Sección 3.2.2 “Estadísticas Acumuladas”. Estos valores (LowThreshold y RangeScaleFactor) serán diferentes para cada plano de color, y solamente necesitan ser calculados una vez por imagen.

Ambas tareas pueden ser emprendidas simultáneamente, como se muestra en la Figura 20.

ES 2 282 136 T3

Puesto que esta etapa incluye un proceso escalar, podemos dejar alguna componente fraccional en el valor mapeado, por ejemplo el valor 12 puede mapearse hasta 5,25. En vez de desechar la componente fraccional, pasamos a un resultado de 10 bits (8 bits de número entero, 2 de fracción) en la siguiente fase de la cadena de procesamiento de imagen. No podemos proporcionar una memoria para almacenar la imagen completa a más de 8 bits, pero podemos hacer un buen uso de la resolución más alta en la fase de remuestreo. Por consiguiente, la imagen de entrada es de 8 bits, y la imagen de salida tiene 10 bits por componente de color. El proceso lógico se muestra en la Figura 21.

Es importante tener un límite inferior de 0 durante la substracción de modo que todos los valores por debajo de LowThreshold 72 serán mapeados hasta 0. De igual modo, la multiplicación debe tener un techo de 255 para la porción entera del resultado, de modo que los valores más altos que HighThreshold 73 serán mapeados hasta 255.

3.2.5 Remuestreo 64

El CFA proporciona solamente un componente de color simple por coordenada (x, y) de píxel. Para producir la imagen impresa final, necesitamos tener los otros valores de componente de color en cada píxel. Finalmente, necesitamos los componentes de color cyan, magenta y amarillo en cada píxel, pero para llegar al cyan, magenta y amarillo necesitamos el rojo, el verde y el azul. Con nuestro un-color-por-píxel, tenemos el componente rojo para una posición particular, pero necesitamos estimar el azul y el verde. O podemos tener el verde y necesitamos estimar el rojo y el azul.

Incluso aunque tuviéramos los componentes de color rojo, verde y azul completos para cada píxel de resolución CFA, la imagen de resolución CFA no es la resolución final de salida. Adicionalmente, aunque el formato de salida varíe, la anchura física de la imagen impresa es constante (10 cm (4 pulgadas) a 1.600 dpi). La anchura constante de la cabeza de impresión es por lo tanto de 6.400 puntos.

Existen dos casos extremos que se deben considerar:

- Interpolarse a resolución CFA (interpolación mínima), y después realizar intensificación, conversión de color. Finalmente, extrapolar hasta la resolución de impresión. Esto tiene la ventaja de una función de intensificación constante y de una conversión de color a baja resolución. Sin embargo, tiene la desventaja de requerir que se almacenen más de 8 bits por componente de color para que los valores de imagen interpolada o intermedia sean interpolados incorrectamente durante la extrapolación final hasta la resolución de impresión. También tiene la desventaja de requerir una unidad de extrapolación que sea capaz de producir 1 valor impresión-res interpolado por ciclo.
- Interpolarse a resolución de impresión, realizando a continuación intensificación y conversión de color. Esto tiene la ventaja de un sólo proceso de remuestreo, proporcionando una máxima precisión. Sin embargo, tiene la desventaja de requerir una unidad de extrapolación que sea capaz de producir 1 valor bi-cúbico interpolado por ciclo, así como también realizar intensificación y conversión de color, todo ello sobre un promedio de un sólo ciclo. La función de intensificación debe ser suficientemente grande para aplicar la función CAFA-res a la imagen de alta-res. Peor aún, para la intensificación, deben mantenerse al menos 3 ventanas sobre la imagen de salida (conteniendo cada una de ellas un número de 6.400 líneas de entrada) puesto que en un ciclo único de impresión, los puntos cyan, magenta y amarillo representan puntos de 6 líneas diferentes.

En ninguno de estos casos se ha tenido en cuenta el hecho de que la salida final de impresión es bi-nivel en vez de contone. En consecuencia, podemos imprimir una imprimación media con relación al remuestreo, y conseguir lo mejor de ambos métodos.

La solución consiste en interpolarse a una resolución intermedia. La intensificación y la conversión de color ocurren a la resolución intermedia, seguido de una extrapolación hasta la resolución de impresión. La resolución intermedia debe ser suficientemente baja como para permitir las ventajas de la contemporización de la función de intensificación de pequeño tamaño y de la conversión de color. Pero la resolución intermedia debe ser suficientemente alta para que no exista pérdida de extrapolación de calidad a la imagen bi-nivel de resolución de impresión. El efecto debe ser la misma que si existiera una interpolación simple a la resolución de impresión (en vez de dos).

Puesto que la imagen de impresión se imprime como 1.600 puntos dpi bi-nivel vibrados, ésta puede ser representada mediante una imagen contone de 320 dpi. En consecuencia, una resolución intermedia de 1.280 píxeles contone proporciona una pérdida no percibida de calidad sobre 6.400 puntos bi-nivel. El último incremento de 1.280 a 6.400 es por lo tanto una relación de extrapolación exacta de 1:5.

Para decidir cómo realizar mejor el remuestreo, es mejor considerar cada plano de color en relación con la resolución CFA. Esto se ha representado en la Figura 22 para un rotacional de 0.

ES 2 282 136 T3

3.2.5.1 Rojo 45 y Azul 47

5 Mirando a los planos rojo 45 y azul 47, la versión completa de resolución CFA del plano de color puede ser creada mediante un escalado del número de píxeles muestreados en cada dimensión por 2. Los píxeles intermedios pueden ser generados mediante un filtro de reconstrucción (tal como un filtro Lanczos o Exponencial). Solamente se requiere una dimensión en la función, puesto que la función es simétrica. Puesto que el rojo y el azul tienen diferentes desviaciones en términos de su representación inicial dentro del espacio de muestra CFA, las posiciones iniciales en la función serán diferentes.

10 El mapeo de las coordenadas de salida (en el espacio de 1.280) respecto a las coordenadas de entrada, depende de la rotación actual de la imagen, debido al registro de los cambios de los píxeles con la rotación (ya sea 0 o ya sea 90 grados dependiendo del formato de impresión). Para el rojo y el azul, se mantiene entonces la siguiente relación:

$$\left. \begin{aligned}
 x' &= \left(\frac{x}{mps} \right) + k_1 \\
 y' &= \left(\frac{y}{mps} \right) + k_2
 \end{aligned} \right\}$$

donde:

x, y = coordenadas en el espacio res medio

x', y' = coordenadas en el espacio de entrada

mps = píxeles de res medio en la muestra espacial

$k_{1,2} = \{0, -0,5\}$ dependiendo de la rotación.

Esto significa que, dada una posición de partida en el espacio de entrada, podemos generar una nueva línea de píxeles de resolución media añadiendo un Δx y un Δy de $1/mps$ y 0 , respectivamente, 1.279 veces. La parte fraccional de x e y en el espacio de entrada, puede ser utilizada directamente para la búsqueda de los coeficientes de función para la construcción y el remuestreo de la imagen.

Obsérvese que k_1 y k_2 son 0 y $-0,5$, dependiendo de si la imagen ha sido girada en 0 o en 90 grados. La Tabla 8 muestra los valores para k_1 y k_2 en los planos del rojo y del azul, suponiendo que la rotación de 90 grados sea anti-horaria.

TABLA 8

Efecto de rotación sobre k_1 y k_2 (la rotación es anti-horaria)

Formato	Rotación Desde CFA Original	Rojo k_1	Rojo k_2	Azul k_1	Azul k_2
Standar 30	90	0	-0,5	-0,5	0
Passport 31	90	0	-0,5	-0,5	0
Panoramic 33	90	0	-0,5	-0,5	0
Thumbnail 32	0	0	0	-0,5	-0,5

El número de píxeles de resolución media por muestra, mps , depende del formato de impresión. Dado que la imagen RGB planarizada tiene las siguientes resoluciones planares roja y azul cuando no está girada: R: 750×500 , B: 750×500 , los factores escalares para los diferentes formatos de salida (Figura 13) se muestran en la Tabla 9. Obsérvese que con el formato de imagen Passport, la imagen completa se remuestrea en $\frac{1}{4}$ del espacio de salida.

TABLA 9

Factores Escalares Rojo y Azul para Formatos de Imagen

Formato	Mapeo	Mps	1/mps
Standard 30	500 ⇒ 1.280	2,56	0,390625
Passport 31	500 ⇒ 640	1,28	0,78125
Panoramic 33	250 ⇒ 1.280	5,12	0,1953125
Thumbnail 32	750 ⇒ 1.280	1,71	0,5848

Según puede apreciarse en la Tabla 9, las imágenes roja y azul están aumentadas para todos los formatos de imagen. En consecuencia, no existirán partículas de solapamiento introducidas por el proceso de remuestreo.

3.2.5.2 Verde 46

El plano 46 verde no puede ser aumentado simplemente de la misma manera que el rojo o el azul, puesto que cada línea del plano verde representa diferentes píxeles, ya sea píxeles impares o ya sean pares sobre líneas alternas. Aunque en términos de número de píxeles sea representativo decir que la imagen verde es de 750 x 1.000, se podría decir igualmente que la imagen es de 1.500 x 500. Esta confusión se debe a la naturaleza de tablero de ajedrez de los píxeles verdes, donde la distancia entre píxeles no es igual en las dimensiones x e y, y no mapean bien respecto a la reconstrucción o remuestreo de imagen. El número de métodos de interpolación utilizados por otros sistemas para la reconstrucción del plano verde, constituye un testimonio respecto a todo esto (desde la replicación próxima más cercana a la interpolación lineal hasta la interpolación bi-lineal y la reconstrucción heurística).

El mapeo de coordenadas de salida (en el espacio de 1.280) respecto a las coordenadas de entrada, es conceptualmente el mismo para el verde que para el rojo y el azul. El mapeo depende de la rotación actual de la imagen., debido al registro de los cambios de los píxeles con la rotación (ya sea 0 o ya sea 90 grados dependiendo del formato de impresión). Para el plano verde se mantiene la siguiente relación:

$$\left. \begin{aligned} x' &= \left(\frac{x}{mps} \right) + k_1 \\ y' &= \left(\frac{y}{mps} \right) + k_2 \end{aligned} \right\}$$

donde:

x, y = coordenadas en el espacio res medio

x', y' = coordenadas en el espacio de entrada

mps = píxeles de res medio en la muestra espacial

k_{1, 2} = {0, -0,5} dependiendo de la rotación

Al igual que con los planos rojo 45 y azul 47, el número de píxeles de resolución media por muestra, mps, depende del formato de impresión. Dado que la imagen RGB planarizada tiene las siguientes resoluciones planares cuando no está girada: R: 750 x 500; B: 750 x 500, G: 750 x 1.000, los factores de incremento para los diferentes formatos de salida (véase la Figura 13) se muestran en la Tabla 10. Obsérvese que con el formato de imagen Passport, la imagen completa se remuestrea en ¼ del espacio de salida.

ES 2 282 136 T3

TABLA 10

Factores Escalares de Plano Verde para Formatos de Imagen

Formato	Mapeo	Mps	1/mps
Standard 30	1.000 ⇒ 1.280	1,28	0,78125
Passport 31	1.000 ⇒ 640	0,64	1,5625
Panoramic 33	500 ⇒ 1.280	2,56	0,390625
Thumbnail 32	1.500 ⇒ 1.280	0,85	1,17648

Estos factores permiten el mapeo de coordenadas entre el espacio de entrada de resolución CCFA y el espacio de resolución media. Sin embargo, una vez que tenemos una coordenada en el espacio de entrada de resolución CFA, no podemos realizar la reconstrucción y el remuestreo de la imagen sobre las muestras de la misma forma que en el rojo y el azul debido a la naturaleza de tablero de ajedrez del plano 46 verde.

Al contrario, a los efectos de reconstrucción de remuestreo de imagen de alta calidad, podemos considerar que el canal verde es una imagen girada en 45 grados. Cuando observamos los píxeles de esta manera, como muestra la Figura 23, aparecen claramente los métodos de reconstrucción y de remuestreo de imagen de alta calidad.

Observando la Figura 23, la distancia entre los píxeles muestreados en las direcciones X e Y es ahora la misma. La distancia real entre los píxeles muestreados es $\sqrt{2}$, según se ilustra en la Figura 24.

La solución para el canal del verde consiste entonces en realizar una reconstrucción y un remuestreo de imagen en el espacio girado. Aunque se utiliza el mismo filtro de reconstrucción que para remuestrear el rojo y el azul, la función deberá ser diferente. Esto se debe a que la relación entre la velocidad de muestreo para el verde y la frecuencia más alta de la señal, es diferente de la relación para los planos rojo y azul. Además, la función deberá ser normalizada de modo que $\sqrt{2}$ de la distancia entre muestras sea igual 1 mientras avanzan las coordenadas de la función (las distancias no normalizadas entre coordenadas de remuestreo deben ser utilizadas aún para determinar si se va a producir solapamiento). Por lo tanto, necesitamos dos transformaciones:

- La primera consiste en mapear el espacio CFA no girado a un espacio CFA girado. Esto puede realizarse multiplicando cada coordenada por $1/\sqrt{2}$, puesto que estamos girando 45 grados ($\cos 45 = \sin 45 = 1/\sqrt{2}$).
- La segunda consiste en aumentar las coordenadas para que se emparejen con la función normalizada, lo que puede realizarse multiplicando cada ordenada por $1/\sqrt{2}$.

Estas dos transformaciones se combinan para crear un factor de multiplicación de $1/2$. Por consiguiente, según avanzamos por el espacio CFA no girado x por k , incrementamos en $k/2$ la función x , y reducimos en $k/2$ la función y . De manera similar, según avanzamos en y por k , incrementamos en $k/2$ la función x e incrementamos en $k/2$ la función y .

Las relaciones entre dos sistemas de coordenadas diferentes pueden ser ilustradas considerando que ocurren según generamos una línea de píxeles de resolución media a partir de una imagen de entrada de espacio CFA. Dada una ordenada de partida en el espacio de entrada CFA, empezamos con $x = 0$, y avanzamos 1.280 veces por $1/\text{mps}$, generando un nuevo píxel en cada nueva posición. El movimiento en el espacio CFA no girado en $1/\text{mps}$, puede ser descompuesto en un movimiento en x y en un movimiento en y en el espacio CFA girado. El proceso se muestra en la Figura 25.

Puesto que $\cos 45 = \sin 45 = 1/\sqrt{2}$, el movimiento en el espacio CFA no girado equivale a igual movimiento en x e y en $1/(\text{mps} \sqrt{2})$. Esta cantidad debe ser aumentada ahora para equipararse con la función normalizada. El incremento equivale a otra multiplicación por $1/\sqrt{2}$. Por consiguiente, un movimiento de $1/\text{mps}$ en el espacio CFA no girado equivale a un movimiento de $1/2$ mps en la función x y en la función y . La Tabla 11 expone la relación entre los tres sistemas coordenados para los diferentes formatos:

TABLA 11

Valores Δ de Función de Plano Verde para Formatos de Imagen

Formato	Factor de Incremento	Δ de Espacio CFA No Girado	Δ de Espacio CFA Girado	Δ de Función
		1	1	1
		mps	$mps\sqrt{2}$	2mps
Standard	1,28	0,78125	0,552	0,391
Passport	0,64	1,5625	1,105	0,781
Panoramic	2,56	0,391	0,276	0,195
Thumbnail	0,85	1,17648	0,832	0,601

La Tabla 11 muestra que el movimiento en el espacio de la función es siempre un número menor de 1, pero en el espacio CFA girado, solamente la imagen Passport tiene un Δ de valor mayor de 1. Como resultado, se producirá solapamiento para el formato de impresión Passport, pero no para ninguno de los otros. Sin embargo, dado que Δ es casi 1, y que cada una de las 4 imágenes es de un tamaño de sólo $\frac{1}{4}$, el solapamiento no será apreciable, especialmente debido a que suponemos un filtro pasa bajo ideal en el verde durante la captura de imagen.

3.2.5.3 Filtro de Reconstrucción para Rojo, Azul y Verde

El filtro de reconstrucción exacta que se va a utilizar, dependerá de un número de circunstancias. Existe siempre una relación entre el número de muestras utilizadas entre la construcción de la señal original, el tiempo invertido en la reconstrucción de la señal, y la calidad de la imagen remuestreada. Una relación satisfactoria en este caso es de 5 muestras de píxel desde la dimensión que está siendo reconstruida, centradas alrededor de la posición X estimada, es decir, X-2, X-1, X, X+1, X+2. Debido a la naturaleza de la reconstrucción con 5 puntos de muestra, solamente necesitamos 4 coeficientes para la introducción de la función de convolución.

Creemos una tabla de búsqueda de coeficiente de función con n entradas para cada componente de color. Cada entrada tiene 4 coeficientes. Según avanzamos en el espacio de salida, mapeamos los cambios en el espacio de salida respecto a los cambios en el espacio de entrada y el espacio de la función. Los bits más significativos de la componente fraccional en el espacio de la función actual, son utilizados para la indexación en la tabla de coeficientes de función. Si existen 64 entradas en la tabla de función, los 6 primeros bits de función son utilizados para buscar los coeficientes, siendo 64 entradas absolutamente suficientes para el remuestreo en la Printcam.

3.2.6 Intensificación

La imagen capturada por el CFA debe ser intensificada con anterioridad a ser imprimida. Idealmente, el filtro de intensificación deberá ser aplicado en el dominio de resolución de CFA. Sin embargo, a la resolución de captura de la imagen, no tenemos información de color completa en cada píxel. Por el contrario, solamente tenemos rojo, azul o verde en una posición de píxel dada. La intensificación de cada plano de color de forma independiente, da lugar a cambios de color. La intensificación deberá ser aplicada, por el contrario, al canal de luminancia de una imagen, de modo que la tonalidad y la saturación de un píxel dado deberán permanecer sin cambio.

La intensificación conlleva entonces la traslación de una imagen RGB a un espacio de color en el que la luminancia esté separada del resto de la información de color 80 (tal como HLS o Lab). El canal 81 de luminancia puede ser entonces intensificado 82 (con la adición en una proporción de la versión filtrada pasa-alto de la luminancia). Finalmente, la imagen completa deberá ser convertida de nuevo en RGB 83 (o en CMY puesto que vamos a imprimirla en CMY). El proceso se muestra en la Figura 26.

Sin embargo, podemos evitar muchas de las etapas de conversión de color si consideramos el efecto de añadir una L filtrada pasa-alto de nuevo a la imagen; el efecto es un cambio en la luminancia de la imagen. Un cambio en la luminancia de un píxel dado puede ser bien aproximado con un cambio igual en R, G y B lineal. Por lo tanto, generamos simplemente L, filtro pasa alto L, y aplicamos una proporción del resultado de igual manera a R, G y B.

ES 2 282 136 T3

3.2.6.1 Convertir RGB en L 80

Consideremos el espacio de color CIE 1976 L*a*b, donde L es perceptiblemente uniforme. Para hacer la conversión de RGB a L (el canal de luminancia), promediamos el mínimo y el máximo de R, G y B, como sigue:

$$L = \frac{\text{MÍN}(R, G, B) + \text{MÁX}(R, G, B)}{2}$$

3.2.6.2. Filtro Pasa Alto L 84

Un filtro pasa alto 84 puede ser aplicado entonces a la información de luminancia. Puesto que estamos filtrando en un espacio med-res de resolución media en vez de un espacio de resolución CFA, el tamaño de la función de intensificación puede ser incrementado o el resultado pasa alto puede ser incrementado apropiadamente. La cantidad exacta de intensificación dependerá del CFA, pero una función 85 de convolución de 3 x 3 será suficiente para producir buenos resultados.

Si tuviéramos que incrementar el tamaño de la función, la Tabla 12 muestra el incremento efectivo requerido para una convolución de 3 x 3 en el espacio CFA según se aplica al espacio de resolución 1.280, utilizando el canal verde como base para el incremento de la función. A partir de esta tabla queda claro que una función dimensionada en 7 x 7, aplicada al espacio de resolución media, será adecuada para toda la intensificación.

TABLA 12

Factores de Incremento para Filtro de Convolución

Formato	Escala	3 x 3	Función en el Espacio Med-res (1.280)
Standard 30	1,28	3,84	3 x 3 ó 5 x 5
Passport 31	0,64	1,92	ninguno, o 3 x 3
Panoramic 33	2,56	7,68	7 x 7
Thumbnail 32	0,85	2,55	ninguno, o 3 x 3

Si se aplicara un filtro 85 de 3 x 3 sobre la imagen med-res, el resultado sería incrementado 86 de acuerdo con el factor de escala utilizado en la operación general de escalado de imagen. Dadas las cantidades de la Tabla 12 (particularmente, el formato de impresión Standard), podemos utilizar un filtro 85 de 3 x 3, y a continuación escalar el resultado. El procedimiento de producción de un píxel L filtrado simple se muestra en la Figura 27.

La función real utilizada puede ser una cualquiera de un conjunto de funciones estándar de filtro pasa alto. Un filtro pasa alto básico pero satisfactorio, se muestra en esta implementación de PCP en la Figura 50.

3.2.6.3 Añadir L Filtrada a RGB

La siguiente cosa que se ha de hacer, consiste en añadir alguna proporción de los valores de luminancia filtrados pasa alto resultantes al canal de luminancia. La imagen puede ser entonces convertida de nuevo en RGB (o en su caso, en CMY). Sin embargo, un cambio de luminancia puede ser aproximada razonablemente mediante un cambio igual en R, G y B (puesto que el espacio de color es lineal). Por consiguiente, podemos evitar la totalidad de conversiones de color mediante la adición de una cantidad igual de valor de luminancia filtrada pasa alto a los R, G y B. La proporción exacta de la imagen filtrada pasa alto puede ser definida por medio de un factor escalar.

Si L es el píxel de luminancia filtrada pasa alto, y k es el factor escalar constante, podemos definir la transformación de intensificación de R, G y B como sigue:

$$\left. \begin{aligned} R' &= R + kL \\ G' &= G + kL \\ B' &= B + kL \end{aligned} \right\} \text{(limitados a 255 cada uno)}$$

Por supuesto, el factor escalar aplicado a L puede ser combinado con el factor escalar del proceso de filtro pasa alto (véase la Sección 3.2.6.2) para obtener un único factor escalar.

ES 2 282 136 T3

Una vez que la intensificación ha sido aplicada al píxel RGB, la imagen puede ser convertida a CMY 83 con el fin de que sea imprimida.

3.2.7 Conversión a CMY 83

En términos teóricos, la conversión de RGB a CMY consiste simplemente en:

$$C = 1 - R$$

$$M = 1 - G$$

$$Y = 1 - B$$

Sin embargo, esta conversión supone que el espacio CMY tiene una respuesta lineal, lo que definitivamente no es cierto en las tintas pigmentadas, y solamente es parcialmente cierto en las tintas a base de tintes. El perfil individual de color de un dispositivo particular (entrada y salida), puede variar considerablemente. Por consiguiente, una conversión precisa, así como para permitir los futuros sensores, tintas e impresoras, se requiere un modelo más preciso para la Printcam.

Las transformaciones requeridas se muestran en la Figura 28. Se elige Lab debido a que es perceptivamente uniforme (a diferencia con XYZ). Con relación al mapeo desde la gama de frecuencia de sensor de imagen hasta la gama de frecuencia de impresora, la gama de frecuencia de impresora está típicamente contenida en su totalidad en la gama de frecuencia de sensor.

En vez de realizar estas transformaciones exhaustivamente, se pueden obtener resultados excelentes mediante conversión tri-lineal basada en 3 conjuntos de tablas de búsqueda en 3D. Las tablas de búsqueda contienen las transformaciones resultantes para la entrada específica según se ha indexado mediante RGB. Se requieren tres tablas: una tabla 90 que mapea RGB a C, una tabla 91 que mapea RGB a M, y una tabla 92 que mapea RGB a Y. Se puede utilizar interpolación tri-lineal para proporcionar el resultado final para aquellas entradas que no estén incluidas en las tablas. El proceso se muestra en la Figura 29.

La interpolación tri-lineal requiere leer 8 valores desde la tabla de búsqueda, y llevar a cabo 7 interpolaciones lineales (4 en la primera dimensión, 2 en la segunda, y 1 en la tercera). Se puede usar alta precisión para los valores intermedios, aunque el valor de salida es de sólo 8 bits.

El tamaño de la tabla de búsqueda requerida depende de la linealidad de las transformaciones. El tamaño recomendado para cada tabla en esta aplicación es de $17 \times 17 \times 17^1$, (con cada 8 bits de entrada. Una tabla de $17 \times 17 \times 17$ tiene 4.913 bytes (menos de 5 KB).

Para indexación en las tablas de 17-por-dimensión, los componentes de color de entrada de 8 bits son tratados como números de punto fijo (4:4). Los cuatro bits de la parte entera dan el índice, y los 4 bits de la fracción se utilizan para la interpolación.

3.2.8 Interpolación Ascendente 67

La imagen CMY de resolución media (1.280 de ancho) debe ser ahora interpolada ascendente hasta la resolución de impresión final (6.400 de ancho).

La relación es exactamente de 1:5 en ambas dimensiones.

Aunque es ciertamente posible interpolar bi-linealmente los 25 valores (1:5 en ambas dimensiones X e Y), los valores resultantes no serán imprimidos contone. Los resultados serán vibrados e imprimidos bi-nivel. Dado que los resultados de 1.600 dpi contone serán transformados en puntos bi-nivel vibrados, la precisión de la interpolación bi-lineal desde 320 dpi a 1.600 dpi no será visible (se eligió la resolución media por este motivo). La replicación de píxel producirá por lo tanto buenos resultados.

La replicación de píxel conlleva simplemente tomar un píxel simple, y utilizarlo como el valor para una gran área. En este caso, replicamos un píxel simple hasta 25 píxeles (un bloque de 5 x 5). Si cada píxel fuera contone, el resultado podría aparecer fraccionado, pero debido a que los píxeles están vibrados, el efecto es que los 25 puntos bi-nivel resultantes adoptan el valor contone. El proceso se muestra en la Figura 30.

1. Aunque una tabla de $17 \times 17 \times 17$ proporcionará excelentes resultados, puede ser posible obtenerlos con una tabla de conversión de $9 \times 9 \times 9$ (729 bytes). El tamaño exacto puede ser determinado por simulación. La aproximación de 5K resultados-conservadores-pero-definitivos fue elegida a efectos de este documento.

ES 2 282 136 T3

3.2.9 *Semitono 68*

La cabeza de impresión 2 es solamente capaz de imprimir puntos de forma bi-nivel. Debemos convertir, por lo tanto, la imagen desde CMY contone a CMY vibrada. Más específicamente, producimos un vibración ordenada de punto disperso utilizando una célula vibratoria estocástica, convirtiendo una imagen CMY contone en una imagen CMY bi-nivel vibrada.

El valor contone de 8 bits de 1.600 dpi, se compara con la posición actual en la célula vibratoria 93. Si el valor contone de 8 bits es mayor que el valor de célula vibratoria, se genera un bit de salida que es un 1. En otro caso, se genera un bit de salida que es un 0. Este bit de salida será enviado eventualmente a la cabeza de impresión y controla una boquilla simple para producir un punto simple C, M o Y. El bit representa si debe dispararse o no una boquilla particular para un color y una posición dados.

La misma posición en la célula 93 vibratoria puede ser utilizada para C, M e Y. Esto se debe a que la cabeza de impresión 2 real produce los puntos C, M e Y para diferentes líneas en el mismo ciclo de impresión. El escalonamiento de los diferentes puntos coloreados proporciona una acumulación en la célula vibratoria.

El proceso de semitono puede ser apreciado en la Figura 31.

El tamaño de la célula 93 vibratoria depende de la resolución de los puntos de salida. Puesto que estamos produciendo puntos de 1.600 dpi, el tamaño de célula deberá ser más grande de 32 x 32. Adicionalmente, para permitir que el orden de procesamiento de punto se empareje con los segmentos de la cabeza de impresión, el tamaño de la célula vibratoria debería dividirse idealmente por 800 de manera uniforme (puesto que existen 800 puntos en cada segmento de la cabeza de impresión).

Un tamaño de célula vibratoria de 50 x 50 es suficientemente grande como para producir resultados de alta calidad, y se divide uniformemente por 800 (16 veces). Cada entrada de la célula vibratoria es de 8 bits, lo que da un total de 2.500 bytes (aproximadamente, 1,5 KB).

3.2.10 *Reformateo para Impresora 69*

El proceso final con anterioridad a ser enviado a la impresora, está destinado a que los puntos sean formateados en el orden correcto con anterioridad a ser enviados a la cabeza de impresión. Los puntos deben ser enviados a la cabeza de impresión en el orden correcto: 24 puntos cada vez como se define en la Sección 2.2.1.

Si los puntos pueden ser producidos en el orden correcto para la impresión (es decir, las funciones vibratoria y de interpolación ascendente generan sus datos en el orden correcto), entonces esos valores de punto (cada valor es 1 bit) pueden ser recopilados de forma simple, y enviados en grupos de 24. El proceso se muestra en la Figura 32.

Los grupos de 24 bits pueden ser enviados a continuación hasta la cabeza de impresión 2 por medio de la Interfaz 15 Memjet.

4. *Núcleo y memoria de CPU*

4.1 *Núcleo de CPU 10*

El PCP 3 incorpora un único núcleo 10 de CPU microcontrolador para sincronizar las cadenas de captura de imagen y de procesamiento de imagen de impresión, y para llevar a cabo las tareas de sistema operativo general de la Printcam, incluyendo la interfaz de usuario. Una amplia variedad de núcleos de CPU son adecuados: pueden consistir en cualquier núcleo procesador con potencia de procesamiento suficiente para realizar las funciones de cálculo y control requeridas de una manera suficientemente rápida como para que se cumplan las expectativas del consumidor.

Puesto que todo el procesamiento de imagen se realiza mediante hardware dedicado, la CPU no tiene que procesar píxeles. Como resultado, la CPU puede ser extremadamente simple. Sin embargo, debe ser suficientemente rápida para hacer correr el motor paso a paso durante una impresión (el motor paso a paso requiere un proceso de 5 KHz). Como ejemplo de núcleo adecuado, se mencionada el microcontrolador Philips 8051 que corre a aproximadamente 1 MHz.

No hay necesidad de mantener continuidad establecida de instrucción entre diferentes modelos de Printcam. Los diferentes diseños de chip de PCP pueden ser fabricados por distintos fabricantes, sin que se requiera licenciar o realizar el núcleo de CPU. Esta independencia de dispositivos evita el monopolio del proveedor de chips como ha ocurrido en el mercado del PC con Intel.

Asociadas al Núcleo de CPU se encuentran una ROM de Programa 13 y una pequeña RAM Programable 14 de Programa.

La CPU 10 comunica con las otras unidades dentro del PCP 3 a través de E/S mapeadas de memoria. La dirección particular establece el mapa de rango respecto a las unidades particulares, y dentro de cada rango, respecto a registros particulares dentro de esa unidad particular. Esto incluye las interfaces serie y paralelo.

ES 2 282 136 T3

4.2 ROM de Programa 13

Una pequeña ROM Flash de Programa 13 se ha incorporado en el PCP 3. El tamaño de la ROM depende de la CPU elegida, pero no será mayor de 16-32 KB.

4.3 RAM de Programa 14

De igual modo, una pequeña área 14 de RAM programable se encuentra incorporada en el PCP 3. Puesto que el código de programa no tiene que manipular imágenes, no hay necesidad de una gran área de trabajo. El tamaño de la RAM depende de la CPU elegida (por ejemplo, mecanismos de apilamiento, sub-rutina de llamada de convenciones, tamaños de registro, etc), pero no deberá ser mayor de alrededor de 4 KB.

4.4 Decodificador de Memoria de CPU 16

El Decodificador 16 de Memoria de CPU es un decodificador simple para satisfacer los accesos de datos de la CPU. El Decodificador traslada las direcciones de datos hacia los accesos de registro de PCP internos por el bus interno de baja velocidad, y por lo tanto habilita las E/S mapeadas de memoria de los registros de PCP.

5. Interfaces de comunicación

5.1 Interfaz 17 de Puerto Serie USB

Éste es un puerto serie USB estándar, conectado al bus 18 de baja velocidad de chip interno. El puerto serie USB está controlado por la CPU 10. El puerto serie permite la transferencia de imágenes hasta, y desde, la Printcam, y permite la impresión DPOF (Digital Print Order Format, Formato de Orden Digital de Impresión) de fotos transferidas bajo control externo.

5.2 Interfaz Serie 19 de Chip QA

Éste consiste en dos puertos serie estándar de baja velocidad, conectados al bus 18 de baja velocidad de chip interno. El protocolo ajustado a CPU entre los dos, se utiliza para autenticar el rollo de impresión [1, 2], y para las siguientes funciones:

- Adquirir las características de la tinta
- Adquirir el volumen de gota recomendado
- Arrastrar la cantidad de papel impreso y requerir nuevo rollo de impresión cuando exista papel insuficiente para imprimir el formato de impresión requerido.

La razón de tener dos puertos consiste en conectar ambos al Chip 4 QA de cámara conectada y la Chip 5 QA del rollo de impresión con la utilización de líneas separadas. Dos chips QA están implementados como Chips de Autenticación [2]. Si solamente se utiliza una sola línea, un fabricante de réplicas de rollos de impresión podría usurpar el mecanismo de autenticación [1].

5.2.1 Chip 5 QA del Rollo de Impresión

Cada consumible de rollo de impresión contiene su propio chip 5 QA. El chip QA contiene información requerida para mantener la mejor calidad de impresión posible, y se implementa con la utilización de un Chip de Autenticación [2]. Los 256 bits de datos se asignan como sigue:

TABLA 13

Los 256 bits del Rollo de Impresión (16)

M[n]	Acceso	Descripción
0	RO ^a	Cabecera Básica, Banderolas, etc (16 bits)
1	RO	Número de serie (16 bits)
2	RO	Número de lote (16 bits)
3	DO ^p	Papel restante en mm (16 bits)
4	RO	Propiedades de la tinta cyan (32 bits)
5	RO	Propiedades de la tinta cyan (32 bits)
6	RO	Propiedades de la tinta magenta (32 bits)

7	RO	Propiedades de la tinta magenta (32 bits)
8	RO	Propiedades de la tinta amarilla (32 bits)
9	RO	Propiedades de la tinta amarilla (32 bits)
10-12	RO	Para expansión futura = 0 (48 bits)
13-15	RO	Bits aleatorios, diferentes en cada chip (48 bits)

- a. Sólo Lectura
b. Sólo Decremento

Antes de cada impresión, se comprueba la cantidad de papel restante por medio de la CPU para asegurar que existe suficiente para el formato de impresión actualmente especificado. Una vez que se ha iniciado cada impresión, la cantidad de papel restante debe ser decrementada en el chip QA del rollo de impresión por medio de la CPU.

5.3 Interfaz Paralelo 6

La interfaz 6 paralelo conecta el PCP 3 con señales eléctricas estáticas individuales. La CPU está capacitada para controlar cada una de estas conexiones como E/S mapeada de memoria por medio del bus de baja velocidad. (Véase la Sección 4.4 para más detalles sobre las E/S mapeadas de memoria).

La Tabla 14 muestra las conexiones con la interfaz paralelo.

TABLA 14

Conexiones con la Interfaz Paralelo

Conexión	Dirección	Patillas
Motor paso a paso de transporte de papel	Hacia Fuera	4
Motor de Guillotina	Hacia Fuera	1
Motor de Enfoque	Hacia Fuera	1
Solenoides de coronación	Hacia Fuera	1
Disparo de flash	Hacia Fuera	1
Excitadores de segmento LCD de estado	Hacia Fuera	7
Excitadores comunes de LCD de estado	Hacia Fuera	4
Sensor de tracción de papel	Hacia Dentro	1
Botones	Hacia Dentro	4
TOTAL		24

5.4 Interfaz 7 JTAG

Una Interfaz 7 JTAG (Joint Test Action Group) estándar ha sido incluida en el PCP 3 a efectos de comprobación. Debido a la complejidad del chip, se requiere una diversidad de técnicas de prueba, incluyendo BIST (Built In Self Test) y aislamiento de bloque funcional. Se supone una porción superior del 10% del área del chip para la circuitería global de comprobación de chip.

6. RAM 11 de imagen

La RAM 11 de imagen se utiliza para almacenar la imagen 42 capturada. La RAM de Imagen es una Flash multi-nivel (2 bits por célula) de modo que la imagen se conserva después de la alimentación haya sido interrumpida.

ES 2 282 136 T3

La cantidad total de memoria requerida para la imagen RGB lineal planarizada es de 1.500.000 bytes (aproximadamente 1,5 MB), organizada como sigue:

- R: $750 \times 500 = 375.000$ bytes
- B: $750 \times 500 = 375.000$ bytes
- G: $750 \times 1.000 = 750.000$ bytes.

La imagen es escrita por la Unidad de Captura de Imagen, y leída tanto por la Unidad 8 de Histograma de Imagen como la Unidad 99 de Generador de Impresión. La CPU 10 no tiene acceso aleatorio directo a la memoria de imagen. Tiene que acceder a los píxeles de imagen a través de la Unidad de Acceso de Imagen.

7. Unidad 12 de captura de imagen

La Unidad de Captura de Imagen contiene toda la funcionalidad requerida por la Cadena de Captura de Imagen, según se describe en la Sección 3.1. La Unidad de Captura de Imagen acepta datos de píxel a través de la Interfaz 98 de Sensor de Imagen, linealiza los datos RGB a través de una tabla 96 de búsqueda, y finalmente escribe la imagen RGB linealizada fuera de la RAM en formato planar. El proceso se muestra en la Figura 33.

7.1 Interfaz 98 de Sensor de Imagen

La Interfaz 98 de Sensor de Imagen (ISI) es una máquina de estado que envía información de control al Sensor de Imagen CMOS, incluyendo pulsos de sincronización de cuadro y pulsos de reloj de píxel, con el fin de leer la imagen. La mayor parte de la ISI es probable que se utilice como célula fuente del fabricante del sensor de imagen. La ISI está controlada en sí mismo por la Máquina de Estado 97 de la Unidad de Captura.

7.1.1 Formato de Sensor de Imagen

Aunque se encuentran disponibles una diversidad de sensores de imagen, solamente vamos a considerar el dispositivo de filtro de color (CFA) Bayer. El CFA Bayer posee un número de atributos que se van a definir ahora.

La imagen capturada por el sensor CMOS (a través de una lente de captación), se supone que ha sido suficientemente filtrada como para eliminar cualquier partícula de solapamiento. El propio sensor tiene una relación de aspecto de 3:2, con una resolución de 1.500 x 1.000 muestras. La disposición de píxel más probable es el conjunto de filtro de color (CFA) Bayer, con cada bloque de 2 x 2 píxeles dispuesto en mosaico 2G como se muestra en la Figura 15.

Cada muestra contiene de R, G o B (correspondiente a rojo, verde y azul, respectivamente), es de 10 bits. Obsérvese que cada píxel del mosaico contiene información acerca de uno sólo de los R, G o B. Las estimaciones sobre la información de color faltante, deben ser realizadas con anterioridad a que la imagen sea imprimida.

Se considera que el CFA realiza alguna cantidad de supresión de ruido patrón fijo (FPN). Se puede requerir alguna supresión AFN adicional.

7.2. Tabla de Búsqueda 96

La tabla de búsqueda 96 es una ROM que mapea el RGB del sensor en un RGB lineal. Ésta equipara el proceso de Linealizar RGB descrito en la Sección 3.1.2. Como tal, la ROM es de 3 KBytes ($3 \times 1.024 \times 8$ bits). 10 bits de dirección provienen de la ISI, mientras que 2 bits de TableSelect son generados por la Máquina de Estado 97 de la Unidad de Captura de Imagen.

7.3 Máquina de Estado 97

La Máquina de Estado 97 de la Unidad de Captura de Imagen genera señales de control para la Interfaz 1 de Sensor de Imagen, y genera direcciones para linealizar el RGB 40 y para planarizar los datos 41 de imagen.

Las señales de control enviadas a la ISI 198 informan a la ISI de que inicie la captura de píxeles, finalice la captura de píxeles, etc.

La dirección de 2 bits enviada a la Tabla de Búsqueda 96 se empareja con la línea actual que está siendo leída desde la ISI. Para líneas pares (0, 2, 4, etc), la dirección de 2 bits es Rojo, Verde, Rojo, Verde, etc. Para líneas impares (1, 3, 5, etc), la dirección de 2 bits es Verde, Azul, Verde, Azul. Esto es cierto con independencia de la orientación de la cámara.

La dirección de 21 bits enviada a la RAM 11 de Imagen, es la dirección escrita para la imagen. Tres registros conservan la dirección actual para cada uno de los planos rojo, verde, y azul. El incremento de las direcciones como píxeles se escribe en cada plano.

7.3.1 Registros

La unidad de Captura de Imagen contiene un determinado número de registros.

TABLA 15

Registros en la Unidad de Captura de Imagen

Nombre	Bits	Descripción
MaxPixels	12	Número de píxeles en cada fila
MaxRows	12	Número de filas de píxeles en la imagen
CurrentPixel	12	Píxel que se está tomando actualmente
CurrentRow	12	Fila que se está procesando actualmente
NextR	21	Dirección en la RAM de Imagen para almacenar el siguiente píxel Rojo. Establece la dirección de inicio del plano rojo con anterioridad a la captura de imagen. Tras la captura de imagen, este registro señalará el byte posterior al plano rojo.
NextG	21	Dirección en la RAM de Imagen para almacenar el siguiente píxel Verde. Establece la dirección de inicio del plano verde con anterioridad a la captura de imagen. Tras la captura de imagen, este registro señalará el byte posterior al plano verde.
NextB	21	Dirección en la RAM de Imagen para almacenar el siguiente píxel Azul. Establece la dirección de inicio del plano azul con anterioridad a la captura de imagen. Tras la captura de imagen, este registro señalará el byte posterior al plano azul.
EvenEven	2	Dirección para el uso de filas pares/ píxeles pares
EvenOdd	2	Dirección para el uso de filas pares/ píxeles impares
OddEven	2	Dirección para el uso de filas impares/ píxeles pares
OddOdd	2	Dirección para el uso de filas impares/ píxeles impares
Go	1	Al escribir aquí un 1, se inicia la captura. Al escribir aquí un 0, se detiene la captura de imagen. Aquí se escribe un 0 automáticamente por medio de la máquina de estado después de que se ha capturado MaxRows de MaxPixels.

Adicionalmente, la Interfaz 98 de Sensor de Imagen contiene un número de registros. Los registros exactos dependerán del Sensor de Imagen 1 elegido.

8. Unidad 9 de acceso de imagen

La Unidad 9 de Acceso de Imagen produce los medios para que la CPU acceda a la imagen en la ImageRAM 11. La CPU puede leer píxeles desde la imagen de la ImageRAM 11 y escribir píxeles de nuevo.

Los píxeles podrían ser leídos a los efectos de almacenamiento de imagen (por ejemplo, a través del USB) 17, o para procesamiento de imagen simple. Los píxeles podrían ser escritos en ImageRAM 11 tras el procesamiento de imagen, como imagen previamente guardada (cargada vía USB), o como imágenes a efectos de patrón de prueba. Los patrones de prueba podrían ser imágenes sintéticas, imágenes de prueba específicas (cargadas a través del USB), o podrían ser valores de disparo de disparo de boquilla de 24 bits para ser cargados directamente en la cabeza de impresión por medio del modo de prueba de la Unidad 99 de Generador de Impresión.

La Unidad 9 de Acceso de Imagen es un mecanismo de acceso directo a la ImageRAM 11, y opera de forma absolutamente simple en términos de 3 registros como se muestra en la Tabla 16.

ES 2 282 136 T3

TABLA 16
Registros IAU

Nombre	Bits	Descripción
ImageAddress	21	Dirección para leer o escribir en ImageRAM
Modo	3	0 = Leer desde ImageRAM en Valor 1 = Escribir Valor en ImageAddress
Valor	8	Valor almacenado en ImageAddress (si Modo = Leer) Valor para almacenar en ImageAddress (si Modo = Escribir)

La estructura de la Unidad de Acceso de Imagen es muy simple, según se muestra en la Figura 35.

La Máquina de Estado 101 simplemente realiza leer/escribir desde/en ImageRAM 11 siempre que la CPU escriba en el registro de Modo.

9. Unidad 8 de histograma de imagen

La Unidad 8 de Histograma de Imagen (IHU), ha sido diseñada para generar histogramas de imágenes según sea requerido por la Cadena de Procesamiento de Imagen de Impresión descrita en la Sección 3.2.2. La IHU solamente genera histogramas para imágenes de formato planar con muestras de 8 bits cada una.

La Unidad 8 de Histograma de Imagen se utiliza típicamente tres veces por impresión. Se acumulan tres histogramas diferentes, uno por cada plano de color. Cada vez que se acumula un histograma, se analizan los resultados con el fin de determinar los umbrales bajo y alto, los factores de escalado, etc., para su uso en el resto del proceso de impresión. Para más información sobre cómo debe ser usado el histograma, véase la Sección 3.2.2.2 y la Sección 3.2.4.

9.1 RAM 102 de Histograma

El propio histograma se almacena en una RAM 102 de 256 entradas, siendo cada entrada de 20 bits. A la RAM de histograma solamente se accede desde dentro de la IHU. Las entradas individuales son leídas y escritas como cantidades de 20 bits.

9.2 Máquina de Estado y Registros 103

La Máquina de Estado 103 sigue el pseudo-código descrito en la Sección 3.2.2.1. La misma está controlada por los registros que se muestran en la Tabla 17.

TABLA 17
Registros en la Unidad de Histograma de Imagen

Nombre	Bits	Descripción
TotalPixels	20	Número de píxeles a contar (decrementos hasta 0)
StartAddress	21	A partir de dónde empezar el conteo
PixelsRemaining	20	Cuantos píxeles quedan por ser contados
PixelValue	8	Una escritura en este registro carga PixelsCount con la entrada de PixelValue desde el histograma
PixelCount	20	Número de píxeles de PixelValue contados en el histograma actual. Es válido tras una escritura en PixelValue
ClearCount	1	Determina si el conteo de histograma debe ser borrado al principio del proceso de histograma. Un 1 hace que los conteos sean borrados, y un 0 hace que los conteos se mantengan intactos (es decir, el siguiente histograma añade los conteos existentes)

Go	1	Escribir aquí un 1 hace que se inicie el proceso de histograma. Escribir aquí un 0 hace que se detenga el proceso de histograma. Aquí se escribe un 0 automáticamente por parte de la máquina de estado después de que TotalPixels ha contado descendentemente hasta 0.
----	---	---

El uso típico de los registros consiste en establecer TotalPixels con el número de píxeles para incluir en el conteo (por ejemplo, 375.000 para el rojo), StartAddress con la dirección del plano rojo, ClearCount con 1, y escribir un 1 en el registro Go. Una vez que el conteo ha finalizado, los valores individuales del histograma pueden ser determinados escribiendo 0-255 en PixelValue y leyendo el Pixelcount correspondiente.

10. Interfaz de cabeza de impresión 105

La Interfaz de Cabeza de Impresión (PHI) 105, es el medio mediante el que el PCP 3 carga la cabeza de impresión 2 Memjet con los puntos que van a ser imprimidos, y controla el proceso real de impresión de punto. El PHI es un reiniciador cíclico lógico para un número de unidades, a saber:

- una Interfaz Memjet (MJI) 15, que transfiere datos hasta la cabeza de impresión Memjet, y controla las secuencias de disparo de boquilla durante una impresión,
- una Unidad de Generador de Impresión (PGU) 99 es una implementación de la mayor parte de la Cadena de Impresión descrita en la Sección 3.2, así como también proporciona un medio de producción de patrones de prueba. La PGU toma una RGB lineal planarizada obtenida a partir de una imagen capturada de formato CFA procedente de la ImageRAM 11, y produce una imagen CMY vibrada de 1.600 dpi en tiempo real según se requiere mediante la Interfaz Memjet 15. Adicionalmente, la PGU tiene un modo de Patrón de Prueba, que permite que la CPU 10 especifique de manera precisa qué boquillas han de dispararse durante una impresión.

Las unidades del interior de la PHI son controladas por un número de registros que son programados mediante la CPU.

La estructura interna de la Interfaz de Cabeza de Impresión ha sido mostrada en la Figura 37.

10.1 Interfaz Memjet 15

La Interfaz Memjet (MJI) 15 conecta el PCP con la cabeza de impresión Memjet externa, proporcionando tanto datos como señales apropiadas para controlar las secuencias de carga y disparo de boquilla durante una impresión.

La Interfaz Memjet 15 es simplemente una Máquina de Estado 106 (véase la Figura 38) que sigue el orden de carga y disparo de la cabeza de impresión que se ha descrito en la Sección 2.2, y que incluye la funcionalidad del ciclo de Precalentamiento y Limpieza según se ha descrito en la Sección 2.4.1 y 2.4.2.

La MJI 15 carga datos en la cabeza de impresión procedentes de una opción entre 2 fuentes de datos:

- Todo 1s. Esto significa que todas las boquillas dispararán durante un ciclo de Impresión posterior, y constituye el mecanismo estándar para cargar la cabeza de impresión para un ciclo de Precalentamiento y Limpieza.
- Desde la entrada de 24 bits mantenida en el registro de Transferencia de la PGU 99. Éste es el medio estándar de imprimir una imagen, si es una foto capturada o un patrón de prueba. El valor de 24 bits procedente de la PGU es enviado directamente a la cabeza de impresión y se envía un pulso de control "Advance" de 1 bit a la PGU. Al final de cada línea, se envía también un pulso "AdvanceLine" de 1 bit hasta la PGU.

La MJI 15 debe ser iniciada después de que la PGU 99 haya preparado ya el primer valor de transferencia de 24 bits. Esto se hace modo que la entrada de datos de 24 bits será válida para la primera transferencia hasta la cabeza de impresión.

La MJI 15 se conecta por lo tanto directamente a la Unidad 99 de Generador de Impresión y a la cabeza de impresión 2 externa. La estructura básica se muestra en la Figura 38.

10.1.1 Conexiones a la Cabeza de Impresión

La MJI 15 tiene las siguientes conexiones con la cabeza de impresión 2, con la detección de entradas y de salidas con respecto a la MJI 15. Los nombres se emparejan con las conexiones de patillas de la cabeza de impresión (véase la Sección 2).

ES 2 282 136 T3

TABLA 18

Conexiones de la Cabeza de Impresión

Nombre	#Patillas	E/S	Descripción
Chromapod-Select	4	S	Seleccionar qué chromapod ha de ser disparado (0-9)
NozzleSelect	4	S	Seleccionar qué boquilla del pod ha de ser disparada (0-9)
Aunable	1	S	Disparar pulso para grupo de fase A
BEnable	1	S	Disparar pulso para grupo de frase B
CDataIn[0-7]	8	S	Salida de cyan para registro de desplazamiento de cyan de los segmentos 0-7
MDataIn[0-7]	8	S	Entrada de magenta para registro de desplazamiento de magenta de los segmentos 0-7
YDataIn[0-7]	8	S	Entrada de amarillo para registro de desplazamiento de amarillo de los segmentos 0-7
SRClock	1	S	Un pulso sobre SRClock (ShiftRegisterClock) carga los valores actuales desde CDataIn[0-7], MDataIn[0-7] e YDataIn[0-7] en los 24 registros de desplazamiento de la cabeza de impresión
Ptransfer	1	S	Transferencia paralelo de datos desde los registros de desplazamiento hasta los bits de NozzleEnable internos de la cabeza de impresión (uno por boquilla)
SenseSeg-Enable	1	S	Un pulso sobre SenseSegEnable ANDed con datos sobre CDataIn[n] selecciona las líneas de detección para el segmento n.
Tsense	1	E	Detección de temperatura
Vsense	1	E	Detección de tensión
Rsense	1	E	Detección de resistividad
Wsense	41	E	Detección de anchura
TOTAL			

10.1.2 Duración de Pulso de Disparo

La duración de los pulsos de disparo por las líneas Aunable y BEnable, depende de la viscosidad de la tinta (la cual depende de la temperatura y de las características de la tinta), y de la cantidad de potencia disponible para la cabeza de impresión. El rango de duración típica de pulso es de 1,3 a 1,8 μ s. La MJI contiene por lo tanto una tabla de duración de pulso programable, indexada por información desde la cabeza de impresión. La tabla de duraciones de pulso permite el uso de una fuente de alimentación de menor coste, y ayuda a mantener una eyección de tinta más precisa.

La tabla de Duración de pulso tiene 256 entradas, y está indexada por los Vsense y Tsense actuales establecidos. Los 4 bits superiores de la dirección proceden de Vsense, y los 4 bits inferiores de la dirección proceden de Tsense. Cada entrada es de 8 bits, y representa un valor de punto fijo en la gama de 0-4 μ s. El proceso de generación de las líneas Aunable y BEnable se muestra en la Figura 39.

La tabla de 256 bytes es escrita por la CPU 10 con anterioridad a la impresión de la foto. Cada entrada de duración de pulso de 8 bits en la tabla, combina:

- Establecimiento de brillos
- Curva de viscosidad de la tinta (desde el Chip QA) 5
- Rsense

- Wsense
- Tsense
- Vsense

10.1.3 *Conteos de Puntos*

La MJI 15 mantiene un conteo del número de puntos de cada color disparados desde la cabeza de impresión 2. El conteo de puntos para cada color es un valor de 32 bits, borrado individualmente bajo control del procesador. Cada conteo de puntos puede mantener un conteo de puntos de máxima cobertura de 69 impresiones de 15 cm (6 pulgadas), aunque en la utilización típica, el conteo de puntos puede ser leído y borrado después de cada impresión.

Mientras que en el producto Printcam inicial, el consumible contiene tanto papel como tinta, resulta concebible que un modelo diferente de Printcam tenga un consumible reemplazable de tinta solamente. El producto de Printcam inicial puede hacer un conteo descendente de la cantidad de milímetros que restan de papel (almacenado en el chip 5 QA, véase la Sección 5.2), para saber si existe suficiente papel disponible para imprimir el formato deseado. Existe tinta suficiente para la cobertura completa del papel suministrado. En el producto Printcam alternativo, los conteos de puntos pueden ser utilizados por la CPU para actualizar el chip 5 QA con el fin de predecir cuándo se va a agotar la tinta del cartucho de tinta. El procesador conoce el volumen de tinta del cartucho para cada uno de C, M e Y a partir del chip 5. El conteo del número de gotas elimina la necesidad de sensores de tinta, y evita que los canales de tinta se queden secos. Un conteo de gotas actualizado se escribe en el chip 5 QA después de cada impresión. Una nueva foto no será imprimida a menos que exista suficiente tinta, y permite al usuario cambiar la tinta sin que se obtenga una foto insatisfactoria que tenga que ser reimprimida.

La disposición del contador de puntos par el cyan, se muestra en la Figura 40. Los 2 contadores de puntos restantes (MDotCount e UDotCount, para magenta y amarillo respectivamente), tienen una estructura idéntica.

10.1.4 *Registros*

La CPU 10 comunica con la MJI a través de un registro establecido. Los registros permiten a la CPU parametrizar una impresión, así como también recibir información acerca del progreso de la impresión.

Los registros que siguen están contenidos en la MJI:

TABLA 19
Registros de Interfaz Memjet

Nombre de Registro	Descripción
Parámetros de Impresión	
NumTransfers	El número de transferencias requeridas para cargar la cabeza de impresión (normalmente 800). Éste es el número de pulsos sobre el SRClock y el número de valores de datos de 24 bits a transferir a una línea dada.
PulseDuration	Número de punto fijo para determinar la duración de un pulso simple sobre las líneas ColorEnable. Rango de duración = 0 – 6 µs
NumLines	El número de ciclos de Carga/ Impresión que se ha de realizar
Monitorización de la Impresión	
Estado	El Registro de Estado de la Interfaz Memjet
LinesRemaining	El número de líneas que falta por imprimir. Solamente es válida mientras Go = 1. El valor de partida es NumLines.
TransfersRemaining	El número de transferencias que faltan antes de la Cabeza de Impresión se considere cargada para la actual línea. Solamente es válido mientras Go = 1
SenseSegment	El valor de 8 bits para disponer sobre las líneas de datos de

	Cyan durante un pulso SenseSegSelect de información posterior. Solamente se podría establecer 1 de los 8 bits, correspondiente a uno de los 8 segmentos.
SetAllNozzles	Si no es cero, el valor de 24 bits escrito para la cabeza de impresión durante el proceso de LoadDots es todo 1s, de modo que todas las boquillas serán disparadas durante el proceso PrintDots posterior. Éste se utiliza durante los ciclos de precalentamiento y de limpieza. Si es 0, el valor de 24 bits escrito para la cabeza de impresión proviene de la Unidad de Generador de Impresión. Éste es el caso durante la impresión real de una foto y de cualquier imagen de prueba.
Acciones	
Reset	Una escritura en este registro, resetea la MJI, detiene cualquier proceso de carga o impresión, y carga todos los registros con 0.
SenseSegSelect	Una escritura en este registro con cualquier valor, borra el bit de Informe del registro de Estado, y envía un pulso por la línea de SenseSegSelect y los bits de estado de LoadDots y de PrintingDots son todos 0. Si se establece cualquiera de los bits de estado, el bit de Información se borra y no hace nada más. Una vez que las diversas líneas de detección han sido comprobadas, los valores se disponen en los registros de Tsense, Vsense, Rsense y Wsense, y a continuación se establece el bit de Información del registro de Estado. La información continúa durante cualesquiera operaciones posteriores de impresión.
Go	Una escritura de 1 en este bit, inicia los ciclos de LoadDots/PrintDots. Se imprime un total de líneas de NumLines, conteniendo cada una de ellas transferencias NumTransfer de 24 bits. Según se imprime cada línea, LinesRemaining se decrementa, y TransfersRemaining se vuelve a cargar con NumTransfers de nuevo. El registro de estado contiene información de estado de impresión. Al completar NumLines, el proceso de carga/ impresión se detiene y el bit Go se borra. Durante el ciclo de impresión final, no se carga nada en la cabeza de impresión. Una escritura de un 0 en este bit, detiene el proceso de impresión, pero no borra ninguno de los otros registros.
ClearCounts	Una escritura en este registro borra los registros CDotCount, MDotCount, e YDotCount, se registra si los bits son 0, 1 ó 2, respectivamente. En consecuencia, la escritura de 0 no tiene ningún efecto.
Informes	
Tsense	Lee solamente información de Tsense procedente del último pulso SenseSegSelect enviado al segmento SenseSegement. Solamente es válido si se establece el bit FeedbackValid del

	registro de Estado.
5	Vsense Lee solamente información de Vsense a partir del último pulso SenseSegSelect enviado al segmento SenseSegment. Solamente es válido si se establece el bit FeedbackValid del registro de Estado.
10	Rsense Lee solamente información de Rsense a partir del último pulso SenseSegSelect enviado al segmento SenseSegment. Solamente es válido si se establece el bit FeedbackValid del registro de Estado
15	Wsense Lee solamente información de Wsense a partir del último pulso SenseSegSelect enviado al segmento SenseSegment. Solamente es válido si se establece el bit FeedbackValid del registro de Estado
20	CDotCount Lee solamente el conteo de 32 bits de puntos cyan enviados a la cabeza de impresión
25	MDotCount Lee solamente el conteo de 32 bits de puntos magenta enviados a la cabeza de impresión
	YDotCount Lee solamente el conteo de 32 bits de puntos amarillo enviados a la cabeza de impresión

El Registro de Estado de la MJJ es un registro de 16 bits con interpretaciones de bit como sigue:

TABLA 20

Registro de Estado de la MJJ

Nombre	Bits	Descripción
40	1	Si se ha establecido, la MJJ está normalmente cargando puntos, manteniendo el número de puntos que han de ser transferidos en TransfersRemaining. Si se borra, la MJJ no carga normalmente puntos.
45	1	Si se ha establecido, la MJJ está normalmente imprimiendo puntos. Si se borra, la MJJ no imprime normalmente puntos.
50	1	Este bit se establece mientras exista un pulso por la línea Aunable.
	1	Este bit se establece mientras exista un pulso por la línea BEnable
55	1	Este bit se establece mientras sean válidos los valores Tsense, Vsense, Rsense y Wsense.
	3	—
60	4	Éste mantiene el chromapod actual que está siendo disparado se encuentre establecido el bit de estado de PrintingDots.
65	4	Éste mantiene la boquilla actual que está siendo disparada mientras se encuentre establecido el bit de estado de PrintingDots.

ES 2 282 136 T3

10.1.5 Ciclos de Precaentamiento y de Limpieza

Los ciclos de Limpieza y de Precaentamiento se realizan de manera simple estableciendo registros apropiados:

- 5 • SetAllNozzles = 1
- Establecer el registro PulseDuration ya sea en una duración baja (en el caso del modo de precaentamiento) o ya sea en una duración apropiada de eyección de gota para el modo de limpieza.
- 10 • Establecer NumLines de modo que sea el número de veces que la boquilla debe ser disparada.
- Establecer el bit Go y esperar a continuación para que el bit Go sea borrado cuando los ciclos de impresión se hayan completado.

15 10.2 Unidad de Generador de Impresión 99

La Unidad de Generador de Impresión (PGU) 99 es una implementación de la mayor parte de la Cadena de Impresión descrita en la Sección 3.2, así como también la provisión de un medio de producción de patrones de prueba.

20 Desde el punto de vista más simple, la PGU proporciona la interfaz entre la RAM 11 de imagen y la Interfaz Memjet 15, como se muestra en la Figura 41. La PGU toma un RGB lineal planarizado obtenido a partir de una imagen capturada de formato CFA procedente de la ImageRAM, y produce una imagen CMY vibrada de 1.600 dpi en tiempo real según requiere la Interfaz Memjet. Adicionalmente, la PGU tiene un modo de Patrón de Prueba, que permite a la CPU especificar de manera precisa qué boquillas se disparan durante una impresión. La MJI 15 proporciona a la PGU
25 99 un pulso Advance una vez que se han usado los 24 bits, y un pulso AdvanceLine al final de una línea.

La PGU 99 tiene 2 cadenas de procesamiento de imagen. La primera, el modo de Patrón de Prueba, lee simplemente datos directamente desde la RAM de Imagen 11, y los formatea en una memoria intermedia lista para proporcionar una salida para la MJI. La segunda contiene la mayoría de las funciones de Cadena de Impresión (véase la Sección
30 3.2). La Cadena de Impresión mostrada en la Figura 18 contiene las funciones:

- Acumular Estadísticas 60
- Girar Imagen 61
- 35 • Balance del Blanco 62
- Expansión de Rango 63
- 40 • Remuestreo 64
- Intensificación 65
- Convertir a CMY 66
- 45 • Interpolación Ascendente 67
- Semitono 68
- 50 • Reformato para Impresora 69.

La PGU 99 contiene todas estas funciones con la excepción de Acumular Estadísticas 60. Para realizar la etapa de Acumular Estadísticas, la CPU 10 llama a la Unidad de Histograma de Imagen 8 tres veces (una vez por canal de color), y aplica algunos algoritmos simples. El resto de las funciones son el dominio de la PGU 99 por razones de
55 precisión y de velocidad: precisión, porque podría ser que se precisara demasiada memoria para mantener la imagen completa con una alta precisión, y velocidad, debido a que una sola CPU 10 no puede mantener las demandas de alta velocidad en tiempo real de la cabeza de impresión 2 Memjet.

La PGU 99 toma como entrada una diversidad de parámetros, incluyendo tablas de conversión de RGB a CMY, constantes para realizar el balance del blanco y la expansión de rango, factores de escala para remuestreo, y parámetros de acceso de imagen que permitan la rotación.
60

Las dos cadenas de proceso pueden verse en la Figura 20. La cadena más directa va desde la RAM de Imagen 11 hasta la Memoria Intermedia 5 por medio del proceso 110 de Acceso de Patrón de Prueba. La otra cadena consiste en 5 procesos, corriendo todos ellos en paralelo. El segundo proceso 112 realiza el Remuestreo. El tercer proceso 65 realiza la intensificación. El cuarto proceso 66 realiza la conversión de color. El proceso 113 final realiza la interpolación ascendente, el semitono y el reformato de la impresora. Los procesos están conectados a través de memorias intermedias, solamente con unos pocos bytes entre algunos procesos, y con unos pocos kilobytes para otros.

ES 2 282 136 T3

Observemos estos procesos y memorias intermedias principalmente en un orden inverso, puesto que la temporización para la cabeza de impresión activa el proceso completo. Las temporizaciones para procesos particulares y los requisitos de tamaño de la memoria intermedia son entonces más evidentes. No obstante, los tamaños de memoria intermedia se muestran en forma de sumario en la Tabla 21.

TABLA 21

Tamaños de memoria intermedia para la Unidad de Generador de Impresión

Memoria Intermedia	Tamaño (bytes)	Composición de la Memoria Intermedia
Memoria Intermedia 1	188	Memoria Intermedia del Rojo = 6 líneas de 6 entradas @ 10 bits cada una = 45 bytes Memoria Intermedia del Azul = 6 líneas de 6 entradas @ 10 bits cada una = 45 bytes Memoria Intermedia del Verde = 13 líneas de 6 entradas @ 10 bits cada una = 97,5 bytes
Memoria Intermedia 2	24	6 x 4 RAM 3 líneas de 4 entradas de L @ 8 bits cada una = 12 bytes 3 colores x 4 entradas @ 8 bits cada una = 12 bytes
Memoria Intermedia 3	3	3 colores (RGB) @ 8 bits cada uno
Memoria Intermedia 4	23.040	3 colores (CMY) x 6 líneas x 1.280 píxeles @ 8 bits cada una
Memoria Intermedia 5	9	3 x 24 bits
TOTAL	23.264	

Aparte del número de registros, algunos de los procesos tienen tablas de búsqueda o componentes de memoria significativas. Éstos se resumen en la Tabla 22.

TABLA 22

Requisitos de memoria dentro de los Procesos de la PGU

Unidad	Tamaño (bytes)	Composición de Requisitos
Girar/ Balance del Blanco/ Expansionar Rango	0	
Remuestrear/ Convertir a L	1.152	3 funciones, cada una de 64 x 4 x 12 bits
Intensificar	0	
Convertir a CMY	14.739	3 tablas de conversión, cada una de 17 x 17 x 17 x 8 bits
Interpolación Ascendente/ Semitono/ Reformato	2.500	Célula Vibratoria, 50 x 50 x 8 bits
Acceso de Patrón de Prueba	0	
TOTAL	18.931	

ES 2 282 136 T3

10.2.1 Acceso de Patrón de Prueba

El proceso 110 de Acceso de Patrón de prueba es el medio mediante el que se producen los patrones de prueba. Bajo circunstancias de usuario normales, este proceso no será utilizado. Está previsto principalmente a efectos de diagnóstico.

El Acceso de Patrón de Prueba 110 lee la RAM de Imagen 11, y pasa los valores de 8 bits directamente a la Memoria Intermedia 5 118 para presentarlos a la salida para la Interfaz Memjet. Éste no modifica en modo alguno los valores de 8 bits. Los datos de la RAM de Imagen 11 podrían ser producidos por la CPU 10 utilizando la Unidad 9 de Acceso de Imagen.

La lectura de datos desde la RAM de Imagen 11 se realiza de una manera de reiniciación cíclica muy simple. Se utilizan dos registros para describir los datos de prueba: la dirección de inicio del primer byte, y el número de bytes. Cuando se alcanza el fin de los datos, los datos se leen de nuevo desde el principio.

La estructura de la Unidad 110 de Acceso de Patrón de Prueba se muestra en la Figura 43.

Según puede apreciarse en la Figura 43, la Unidad 110 de Acceso de Patrón de Prueba es poco más que el Generador 119 de Dirección. Cuando se inicia, y con cada señal AdvanceLine, el generador lee 3 bytes, produce un pulso TransferWriteEnable, lee los 3 bytes siguientes, y a continuación espera por un pulso de Advance. En el pulso Advance, se proporciona el pulso de TransferWriteEnable, se leen los 3 bytes siguientes, y se produce de nuevo la espera. Esto continúa hasta el pulso AdvanceLine, con el que comienza de nuevo el proceso desde la dirección actual.

En términos de lectura de 3 bytes, el Generador 119 de Dirección simplemente lee tres valores de 8 bits desde la RAM de Imagen 11, y los escribe en la memoria Intermedia 5 118. El valor de los primeros 8 bits se escribe en la dirección 0 de 8 bits de la Memoria Intermedia 5, el siguiente se escribe en la dirección 1 de 8 bits de la Memoria Intermedia 5, y el tercero se escribe en la dirección 2 de 8 bits de la Memoria Intermedia 5. El Generador 110 de Dirección espera después un pulso Advance antes de hacer la misma cosa nuevamente.

Las direcciones generadas para la RAM de Imagen 11, están basadas en una dirección de inicio y en un conteo de bytes como se muestra en la Tabla 23.

TABLA 23

Registros de Acceso de Patrón de Prueba

Nombre de Registro	Descripción
TestModeEnabled	Si es 1, TestMode está habilitado Si es 0, TestMode no está habilitado
DataStart	Dirección de Inicio de los datos de prueba en la RAM de Imagen
DataLength	Número de 3 bytes en los datos de prueba

El siguiente pseudo-código ilustra la generación de dirección. Los pulsos AdvanceLine y Advance no se muestran.

```

Hacer Siempre
    Adr = DataStart
    Resto = DataLength
    Leer Adr en Memoria Intermedia 5 (0), Adr = Adr + 1
    Leer Adr en Memoria Intermedia 5 (1), Adr = Adr + 1
    Leer Adr en memoria Intermedia 5 (2), Adr = Adr + 1
    Resto = Resto - 1
    Si (Resto = 0)
        Resto = Datalength
EndDo
    
```

ES 2 282 136 T3

Es responsabilidad de la CPU 10 asegurar que los datos tienen significado para la cabeza de impresión 2. El Byte 0 es el dato de disparo de boquilla para los 8 segmentos de cian (bit 0 = segmento 0, etc), el Byte 1 es lo mismo para el magenta, y el Byte 2 para el amarillo. Conjuntos alternos de 24 bits se destinan a los píxeles impares/ pares separados por 1 línea horizontal de puntos.

10.2.2 Memoria Intermedia 5 118

La memoria Intermedia 5 118 mantiene los puntos generados a partir del proceso completo de Generación de Impresión. La memoria Intermedia 5 consiste en un registro de desplazamiento de 24 bits para mantener los puntos generados, uno cada vez, desde la UHRH 113 (Unidad de Interpolación Ascendente-Semitono y Reformateo), 3 registros de 8 bits para mantener los datos generados desde la TPAU (Unidad de Acceso de Patrón de Prueba), y un registro de 24 bits utilizado como memoria intermedia para transferir datos a la MJI (Interfaz Memjet). El pulso Advance procedente de la MJI carga el registro Transfer de 24 bits con los 24 bits, ya sea desde los 3 registros de 8 bits o ya sea desde el registro de desplazamiento único de 24 bits.

La Memoria Intermedia 5 actúa por lo tanto como doble mecanismo de almacenamiento intermedio para los puntos generados, y tiene la estructura que se muestra en la Figura 44.

10.2.3 Memoria Intermedia 4 117

La Memoria Intermedia 4 117 mantiene la imagen contone de resolución intermedia CMY calculada (1.280-res). La memoria Intermedia 4 se genera mediante el proceso 66 de Conversión de Color, y es accesible para el proceso 113 de Interpolación-Ascendente, Semitono y Reformateo con el fin de generar puntos de salida para la impresora.

El tamaño de la Memoria Intermedia Contone, depende de la distancia física entre las boquillas de la cabeza de impresión. Según están siendo generados los puntos de un color para una línea física, están siendo generados los puntos de un color diferente por una línea diferente. El efecto neto es que se imprimen 6 líneas físicas diferentes de una vez desde la impresora: puntos impares y pares desde diferentes líneas de salida, y líneas diferentes por color. Este concepto ha sido explicado, y las distancias se han definido en la Sección 2.1.1.

La conclusión práctica es que existe una distancia dada en puntos de alta resolución desde los puntos cian pares a través de los puntos magenta hasta los puntos amarillos impares. Con el fin de minimizar la generación de RGB y con ello CMY, los píxeles contone de resolución media se almacenan temporalmente en la Memoria Intermedia 4.

Puesto que la relación de líneas de media resolución respecto a líneas de alta resolución es de 1:5, cada línea de media resolución se muestrea 5 veces en cada dimensión. A los efectos de líneas de memoria intermedia, solamente estamos afectados por las de 1 dimensión, de modo que sólo consideraremos 5 líneas de puntos procedentes de una sola línea de píxel. La distancia entre boquillas de diferentes colores es de 4-8 puntos (dependiendo de los parámetros de la Memjet). Por lo tanto suponemos 8, que proporciona una distancia e separación de 16 puntos, o 17 puntos en la distancia inclusiva. El peor caso lo constituye un escenario en el que la línea de 17 puntos de línea incluye la última línea de puntos procedente de una línea de píxel dada. Esto implica 5 líneas de píxel, con líneas de puntos generadas como 1, 5, 5, 5, 1, y permite un incremento de separación de boquilla hasta 10.

Para asegurar que el proceso de generación de contone que escribe en la memoria intermedia no interfiere con el proceso de generación de puntos que se lee desde la memoria intermedia, hemos añadido una línea extra de media resolución por color, para un total de 6 líneas por color.

La memoria intermedia contone tiene por lo tanto 3 colores por 6 líneas, conteniendo cada una de las líneas 1.280 valores contone de 8 bits. La memoria total requerida es de $3 \times 6 \times 1.280 = 23.040$ bytes (22,5 KBytes). La memoria solamente requiere una lectura única de 8 bits por ciclo, y una escritura única de 8 bits por cada 25 ciclos (cada píxel contone se lee 25 veces). La estructura de Memoria Intermedia 4 se muestra en la Figura 45.

La memoria Intermedia 4 puede ser implementada como una RAM de ciclo único de doble acceso (lectura y escritura) que corre a la velocidad normal del proceso de generación de puntos de cabeza de impresión, o puede ser implementada como RAM que corre un 4% más rápida con un sólo acceso de lectura o de escritura por ciclo.

La Memoria Intermedia 4 se establece en el blanco (todo 0) con anterioridad al inicio del proceso de impresión.

10.2.4 Interpolación Ascendente, Semitono y Reformateo para Impresora

Aunque las tareas 113 de Interpolación Ascendente, Semitono y Reformateo se definen como tareas separadas por medio de la Sección 3.2.8, la Sección 3.2.9 y la Sección 3.2.10 respectivamente, éstas se implementan como un proceso único en la implementación de hardware del PCP 3.

La entrada a la Unidad 113 de Interpolación Ascendente, Semitono y Reformateo (UHRU), es la memoria intermedia 117 contone (Memoria Intermedia 4) que contiene la imagen CMY de resolución 1.280 (resolución intermedia) calculada previamente. La salida es un conjunto de valores de 24 bits en el orden correcto en que han de ser enviados a la Interfaz Memjet 15 para su posterior salida para la cabeza de impresión por medio de la memoria Intermedia 5

ES 2 282 136 T3

118. Los 24 bits de salida se generan a razón de 1 bit cada vez, y son enviados al registro de desplazamiento de 24 bits de la memoria Intermedia 5 118.

5 El control de este proceso ocurre a partir de las señales Advance y AdvanceLine procedentes de la MJI 15. Cuando se inicia la UHRU 13, y después de cada pulso AdvanceLine, se producen 24 bits, y son cronometrados en el registro de desplazamiento de 24 bits de la Memoria Intermedia 5 por medio de una señal ShiftWriteEnable. Una vez que el bit 24º ha sido cronometrado, se proporciona un pulso TransferWriteEnable y se generan los siguientes 24 bits. Después de esto, la UHRU 113 espera por el pulso Advance procedente de la MJI. Cuando llega el pulso Advance, se proporciona el pulso TransferWriteEnable a la Memoria Intermedia 5 118, y se calculan los siguientes 24 bits con anterioridad a una nueva espera. En la práctica, una vez que se ha proporcionado el primer pulso Advance, se produce la sincronización y los futuros pulsos Advance ocurrirán cada 24 ciclos posteriores.

El proceso de Interpolación Ascendente, Semitono y Reformato puede ser apreciado en la Figura 46.

15 La tarea de Semitono se acomete mediante el comparador simple 120 sin signo de 8 bits. Las dos entradas hacia el comparador proceden de la Célula 121 Vibratoria Acumulada y de la Memoria Intermedia 4 117. El orden en que estos valores son presentados al comparador 120 Sin Signo, se determina mediante la Máquina de Estado 122 de Generador de Dirección, la cual asegura que las direcciones en la imagen de resolución 1.280 se emparejan con el orden de segmento orientado requerido para la cabeza de impresión. La Máquina de Estado 122 de Generador de Dirección acomete por lo tanto las tareas de Interpolación Ascendente y Reformato para la Impresora. En vez de que simplemente acceda una línea completa cada vez a alta resolución, y reformatear después la línea de acuerdo con los requisitos de búsqueda de impresora (como se ha descrito en la Sección 3.2.10), el reformato se logra mediante el direccionamiento apropiado de la memoria intermedia 117 de contone (Memoria Intermedia 4), y asegurando que el comparador 120 utiliza la búsqueda correcta a partir de la célula 121 vibratoria para emparejar las direcciones acumuladas.

30 La tarea de Semitono es la misma que se ha descrito en la Sección 3.2.9. Sin embargo, puesto que las salidas de puntos se generan en el orden correcto para la cabeza de impresión, el tamaño de la Célula Vibratoria 121 se elige de modo que se divide uniformemente en 800. Por consiguiente, una posición dada de la célula vibratoria para un segmento será la misma para los 7 segmentos restantes. Una célula vibratoria de 50 x 50 proporciona un resultado satisfactorio. Según se ha descrito en la Sección 3.2.9, la misma posición de la célula vibratoria puede ser utilizada para diferentes colores debido al hecho de que las diferentes líneas están siendo generadas al mismo tiempo para cada uno de los colores. El direccionamiento para la célula vibratoria es por lo tanto muy simple. Comenzamos en una fila particular de la célula Vibratoria Acumulada (por ejemplo, la fila 0). La primera entrada de célula utilizada es la Entrada 0. Utilizamos esa entrada 24 veces (24 ciclos) para generar 3 colores para los 8 segmentos, y a continuación avanzamos hasta la Entrada 1 de la fila 0. Después de la Entrada 49, volvemos de nuevo a la Entrada 0. Esto continúa para la totalidad de los 19.200 ciclos con el fin de generar 19.200 puntos. La Unidad de Semitono se interrumpe a continuación y espera por el pulso AdvanceLine que provoca que el generador de dirección avance hasta la siguiente fila de la célula vibratoria.

40 La célula 121 Vibratoria Acumulada se denomina así debido a que difiere de una célula vibratoria regular al tener las líneas impares y pares acumuladas. Esto se debe a que generamos píxeles impares y pares (empezando por el píxel 0) sobre diferentes líneas, y se evita que el Generador de Dirección 122 tenga que avanzar hasta la siguiente fila y volver de nuevo sobre conjuntos alternativos de 24 píxeles. La Figura 25 muestra una célula 93 vibratoria única, y cómo mapearla en una célula 121 vibratoria acumulada del mismo tamaño. Obsérvese que para determinar la “condición de impar” de una posición dada, numeramos los píxeles de una fila dada como 0, 1, 2, etc.

50 El valor de 8 bits procedente de la Memoria Intermedia 4 117, se compara (sin signo) con el valor de 8 bits procedente de la Célula Vibratoria Acumulada 121. Si el valor de píxel de la Memoria Intermedia 4 es mayor, o igual, que el valor de célula vibratoria, se dispone a la salida un bit “1” para el registro de desplazamiento de la Memoria Intermedia 5 118. En otro caso, se dispone a la salida un bit “0” para el registro de desplazamiento de la Memoria Intermedia 5.

55 Con relación a los 19.200 píxeles contone de semitono, se deben leer los 19.200 píxeles contone. La Unidad 122 de Generador de Dirección realiza esta tarea, generando las direcciones en la Memoria Intermedia 4 117, implementando de forma efectiva la tarea UpInterpolate. La generación de dirección para lectura de la Memoria Intermedia 4, es ligeramente más complicada que la generación de dirección para la célula vibratoria, pero no excesivamente.

60 El Generador de Dirección para leer la Memoria Intermedia 4, empieza solamente una vez la primera línea de la Memoria Intermedia 4 que ha sido escrita. El resto de filas de la Memoria Intermedia 4 son 0, de modo que estarán efectivamente en blanco (sin puntos imprimidos).

65 Cada una de las 6 líneas efectivas de salida tiene un registro con un componente entero y fraccional. La porción entera del registro se utiliza para seleccionar qué línea de Memoria Intermedia deberá ser leída para la interpolación ascendente efectiva del color para esos píxeles impares y pares del color particular. Se utilizan 3 contra-píxeles para mantener la posición actual dentro del segmento 0, y se utiliza un único contador temporal P_ADR (dirección de píxel) para la desviación hacia los 7 segmentos restantes.

ES 2 282 136 T3

En resumen, la generación de dirección para la Memoria Intermedia 4 requiere los siguientes registros, como se muestra en la Tabla 24.

TABLA 24

Registros Requeridos para la Memoria Intermedia 4 de Lectura

Nombre de Registro	Tamaño
CyanEven	6 bits (3:3)
CyanOdd	6 bits (3:3)
MagentaEven	6 bits (3:3)
MagentaOdd	6 bits (3:3)
YellowEven	6 bits (3:3)
YellowOdd	6 bits (3:3)
Cyan_P_ADR	14 bits (11:3)
Magenta_P_ADR	14 bits (11:3)
Yellow_P_ADR	14 bits (11:3)
P_ADR	11 bits (solamente mantiene la porción entera de X_P_ADR)

Los valores iniciales para los 6 registros de línea de memoria intermedia, son la distancia física entre boquillas (recuérdese que la componente fraccional es efectivamente una parte divisoria por 5). Por ejemplo, si los puntos de salida impares y pares de un color están separados por una distancia de 1 punto, y las boquillas de un color están separadas de las boquillas del siguiente por 8 puntos, los valores iniciales podrían ser como se muestra en la columna "Primera Línea" de la Tabla 25. Una vez que cada conjunto de 19.200 puntos ha sido generado, cada uno de estos contadores debe incrementarse en 1 componente fraccional, que representa el hecho de que estamos muestreando cada píxel 5 veces en la dimensión vertical. Los valores resultantes serán entonces como se muestra en la columna "Segunda Línea" de la Tabla 25. Obsérvese que $5:4 + 1 = 0:0$ puesto que existen solamente 6 líneas de memoria intermedia.

TABLA 25

Ejemplo de Estructura Inicial y de Segunda Línea para los 6 Registros de Línea de Memoria Intermedia

Nombre	Cálculo	Primera Línea		Segunda Línea	
		Valor	Memor.	Valor	Memor.
CyanEven	Posición Inicial	0:0	0	0:1	0
CyanOdd	CyanEven + 0:1	0:1	0	0:2	0
MagentaEven	CyanOdd + 1:3 (8)	1:4	1	2:0	2
MagentaOdd	MagentaEven + 0:1	2:0	2	2:1	2
YellowEven	MagentaOdd + 1:3(8)	3:3	3	3:4	3
YellowOdd	YellowEven + 0:1	3:4	3	4:0	4

Los 6 registros de línea de memoria intermedia determinan entonces cuál de las líneas de memoria intermedia debe ser leída para los píxeles impares o pares de un color dado. Para determina cuales de los 1.280 píxeles de resolución media deben ser leídos desde la línea específica de la Memoria Intermedia 4, utilizamos 3 contadores de Dirección de Píxel, uno por cada color, y un solo contador temporal (P_ADR) que se utiliza para la indexación en cada segmento. Cada segmento está separado del siguiente por 800 puntos. En píxeles de resolución media, esta distancia es de 160. Puesto que 800 es visible exactamente por 5, solamente necesitamos utilizar la porción entera de los 3 contadores de Dirección de Píxel. Generamos las 8 direcciones para los píxeles cyan pares, a continuación las 8 direcciones para los de magenta pares, y finalmente las 8 direcciones para los de amarillo pares. A continuación hacemos lo mismo para los píxeles impares de cyan, magenta y amarillo. Este proceso de dos conjuntos de 24 bits, 24 pares y después 24 impares, se realiza 400 veces. A continuación reseteamos los contadores de Dirección de Píxel (X_P_ADR) a 0, y avanzamos los 6 registros de línea de memoria intermedia. Cada 5 avances de línea, la siguiente línea de memoria intermedia está ahora libre y lista para la actualización (mediante el proceso de Convertir en CMY). La Tabla 26 relaciona las etapas de una manera simple.

TABLA 26

Generación de Dirección para Leer la Memoria Intermedia 4

#	Dirección	Cálculo	Comentario
		P_ADR = Cyan_P_ADR Cyan_P_ADR+ = 1 (mod5)	Generar dirección para el píxel par en el segmento 0 de Cyan y avanzar hasta el siguiente píxel para cyan
1	CyanEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 1 (cyan)
2	CyanEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 2 (cyan)
3	CyanEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 3 (cyan)
4	CyanEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 4 (cyan)
5	CyanEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 5 (cyan)
6	CyanEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 6 (cyan)
7	CyanEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 7 (cyan)
8	CyanEven:P_ADR	P_ADR = Magenta_P_ADR Magenta_P_ADR+=1 (mod5)	Generar dirección para el píxel par en el segmento 0 de Magenta y avanzar hasta el siguiente píxel para magenta.
9	MagentaEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 1 (magenta)
10	MagentaEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 2 (magenta)
11	MagentaEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 3 (magenta)
12	MagentaEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 4 (magenta)
13	MagentaEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 5 (magenta)
14	MagentaEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 6 (magenta)
15	MagentaEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 7 (magenta)
16	MagentaEven:P_ADR	P_ADR = Yellow_P_ADR Yellow_P_ADR+ = 1 (mod5)	Generar dirección para píxel par en segmento 0 de Amarillo, y avanzar hasta el siguiente píxel para el amarillo
17	YellowEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 1 (amarillo)
18	YellowEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 2 (amarillo)
19	YellowEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 3 (amarillo)
20	YellowEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 4 (amarillo)
21	YellowEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 5 (amarillo)
22	YellowEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 6 amarillo
23	YellowEven:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 7 (amarillo)
24	TellowEven:P_ADR	P_ADR = Cyan_P_ADR Cyan_P_ADR+ = 1 (mod5)	Generar dirección para píxel par en segmento 0 de Cyan y avanzar hasta el siguiente píxel para cyan
25	CyanOdd:P_ADR	P_ADR+ = 160	Avanzar hasta el segmento 1 (cyan)
etc.			

El pseudo-código para generar las direcciones de la Memoria Intermedia 4 117, se muestra a continuación. Obsérvese que se ha relacionado como un conjunto secuencial de etapas. La Tabla 26 muestra una visión mejor de la naturaleza paralelo de las operaciones durante la generación de dirección.

```
5  % Calcular posiciones de partida
   CyanEven = 0:0
   CyanOdd = CyanEven + 0:1
   MagentaEven = CyanOdd + 1:3
10  MagentaOdd = MagentaEven + 0:1
   YellowEven = MagentaOdd + 1:3
   YellowOdd = YellowEven + 0:1
15
   Hacerlo N veces (depende del tamaño de impresión)
   Cyan_P_ADR = 0
   Magenta_P_ADR = 0
   Yellow_P_ADR = 0
25  Hacerlo 400 veces
   % generar los píxeles pares para el primer conjunto de 24 bits
   P_ADR = Porción entera de Cyan_P_ADR
   Cyan_P_ADR += 0:1
   Hacerlo 8 veces
   ReadBuffer4 (línea = CyanEven, píxel = P_ADR)
35  P_ADR += 160
   EndDo
   P_ADR = Porción entera de Magenta_P_ADR
   Magenta_P_Adr += 0:1
   Hacerlo 8 veces
   ReadBuffer4 (línea = MagentaEven, píxel = P_ADR)
45  P_ADR += 160
   EndDo
   P_ADR = Porción entera de Yellow_P_ADR
   Yellow_P_Adr += 0:1
   Hacerlo 8 veces
55  ReadBuffer4 (línea = YellowEven, píxel = P_ADR)
   P_ADR += 160
   EndDo
60
65
```

```

% generar los píxeles impares para el primer conjunto de 24
bits
5      P_ADR = Porción entera de Cyan_P_ADR
      Cyan_P_ADR += 0:1
      Hacerlo 8 veces
10         ReadBuffer4 (línea = CyanOdd, píxel = P_ADR)
          P_ADR += 160
      EndDo
15      P_ADR = Porción entera de Magenta_P_ADR
      Magenta_P_Adr += 0:1
      Hacerlo 8 veces
20         ReadBuffer4 (línea = MagentaOdd, píxel = P_ADR)
          P_ADR += 160
      EndDo
25      P_ADR = Porción entera de Yellow_P_ADR
      Yellow_P_Adr += 0:1
      Hacerlo 8 veces
30         ReadBuffer4 ( línea = YellowOdd, píxel = P_ADR)
          P_ADR += 160
35      EndDo

% Ahora puede avanzar hasta la siguiente "línea"
40      CyanEven += 0:1
      CyanOdd += 0:1
45      MagentaEven += 0:1
      MagentaOdd += 0:1
      YellowEven += 0:1
50      YellowOdd += 0:1
      EndDo
      EndDo
55

```

10.2.5 Memoria Intermedia 3 116

60 La memoria Intermedia 3 es un conjunto directo de valores R, G, B de 8 bits. Estos valores RGB son píxeles intensificados de resolución media (resolución de 1.280), generados por medio del proceso 65 de Intensificación, y leídos por el proceso 66 de Convertir en CMY.

65 No es necesario doblar la Memoria Intermedia 116 de almacenamiento temporal. Esto se debe a que el proceso 66 de lectura (Convertir en CMY) solamente requiere los valores RGB para los primeros 39 ciclos, mientras que el proceso 65 de escritura (Intensificación) necesita 49 ciclos antes de ser leídos para actualizar realmente los valores RGB.

ES 2 282 136 T3

10.2.6 Convertir en CMY 66

La conversión de RGB a CMY se realiza en el espacio de resolución medio (resolución de 1.280) como se ha descrito en la Sección 3.2.7.

5 El proceso 66 de conversión debe producir los píxeles contone 117 de almacenamiento intermedio (Memoria Intermedia 4) a una velocidad suficientemente rápida como para seguir el ritmo del proceso 113 de Interpolación Ascendente-Semitono-Reformateo. Puesto que cada valor contone se utiliza durante 25 ciclos (5 veces en cada una de las dimensiones x e y), el proceso de conversión puede necesitar hasta 25 ciclos. Esto totaliza 75 ciclos para el conjunto de las 3 componentes de color.

15 El procedimiento según ha sido aquí descrito, requiere solamente 14 ciclos por componente de color, con los valores RGB de entrada realmente liberados después de 39 ciclos. Si el proceso se implementa con lógica que requiera acceso a los valores RGB de entrada durante más de 49 ciclos, entonces la Memoria Intermedia 3 116 requerirá un doble almacenamiento intermedio, puesto que son actualizados mediante el proceso 65 de Intensificación después de ese tiempo.

20 La conversión se realiza como interpolación tri-lineal. Se utilizan tres tablas de búsqueda de 17 x 17 x 17 para el proceso de conversión: RGB a Cyan 90, RGB a Magenta 91 y RGB a Amarillo 92. Sin embargo, puesto que tenemos que realizar 25 ciclos para efectuar cada interpolación tri-lineal, no hay necesidad de ninguna unidad de interpolación tri-lineal rápida. Al contrario, 8 llamadas a un proceso 130 de interpolación lineal es más que adecuado.

25 La generación de dirección para su indexación en las tablas de búsqueda, es directa. Utilizamos los 4 bits más significativos de cada componente de color de 8 bits para la generación de dirección, y los 4 bits menos significativos de cada componente de color de 8 bits para la interpolación entre valores recuperados desde las tablas de conversión. El direccionamiento en la tabla de búsqueda requiere un sumador debido al hecho de que la tabla de búsqueda tiene dimensiones de 17 en vez de 16. Afortunadamente, la multiplicación de un número X de 4 bits por 17 da como resultado un número XX de 8 bits, y por lo tanto no requiere ningún sumador o multiplicador, y la multiplicación de un número de 4 bits por 17^2 (289) es sólo ligeramente más complicado, requiriendo una única adición.

30 Aunque la interpolación podría ser realizada de forma más rápida, utilizamos un simple sumador para generar direcciones y disponer de una unidad de interpolación de un solo ciclo. Por consiguiente, estamos capacitados para calcular la interpolación para la generación de una única componente de color desde el RGB en 14 ciclos, como se muestra en la Tabla 27. El proceso debe ser repetido 3 veces con el fin de generar cyan, magenta y amarillo. Son posibles métodos más rápidos, pero no es necesario.

TABLA 27

Interpolación tri-lineal para conversión de color

Ciclo	Carga	Búsqueda Efectiva	Ajustar registro de ADR	Interpolación
1			ADR = 289R	
2			ADR = ADR + 17G	
3			ADR = ADR + B	
4	P1	RGB	ADR = ADR + 1	
5	P2	RGB + 1	ADR = ADR + 16	
6	P1	RG + 1B	ADR = ADR + 1	P3 = P1 a P2 Por B
7	P2	RG + 1B + 1	ADR = ADR + 271	
8	P1	R + 1GB	ADR = ADR + 1	P4 = P1 a P2 por B
9	P2	R + 1GB + 1	ADR = ADR + 16	P5 = P3 a P4 por G
10	P1	R + 1G + 1B	ADR = ADR + 1	P3 = P1 a P2 por B
11	P2	R + 1G + 1B + 1		
12				P4 = P1 a P2 por B
13				P6 = P3 a P4 por G
14				V = P5 a P6 por R

ES 2 282 136 T3

Según se muestra en la Tabla 27, se puede utilizar un solo registro ADR y sumador para la generación de dirección en las tablas de búsqueda. Se pueden utilizar 6 conjuntos de registros de 8 bits para mantener los resultados intermedios: 2 registros mantienen los valores cargados desde las tablas de búsqueda, y 4 registros se utilizan para salida desde la unidad de interpolación. Obsérvese que la entrada a la unidad de interpolación lineal consiste siempre en un par de registros de 8 bits P1/P2, P3/P4, y P5/P6. Esto se hace deliberadamente para reducir la lógica de selección de registro. En el ciclo 14, el registro “V” 131 mantiene el valor de 8 bits finalmente calculado. El resultado de 8 bits puede ser escrito en la posición apropiada en la Memoria Intermedia 4 117 durante el siguiente ciclo.

Un diagrama de bloques del proceso 66 de Conversión a CMY, puede ser apreciado en la Figura 48.

Suponiendo que se ejecuta en primer el proceso para generar cyan, el píxel contone cyan resultante se almacena en la memoria intermedia contone de cyan de resolución 1.280. El proceso se ejecuta nuevamente a continuación sobre la misma entrada RGB para generar el píxel magenta. Este píxel magenta contone se almacena en la memoria intermedia contone de magenta de resolución 1.280. Finalmente, el píxel contone amarillo se genera a partir de la misma entrada RGB, y el píxel resultante amarillo se almacena en la memoria intermedia contone de amarillo de resolución 1.280.

La generación de dirección para escribir en la memoria intermedia contone 117 (Memoria Intermedia 4), es directa. Se utiliza una sola dirección (y los bits ColorSelect que la acompañan) para escribir en cada una de las tres memorias intermedias de color. La memoria intermedia de Cyan es escrita en el ciclo 15, la Magenta en el ciclo 30, y la Amarilla en el ciclo 45. La dirección de píxel se incrementa en 1 cada 75 ciclos (después de que la totalidad de los 3 colores han sido escritos). La línea en la que se está escribiendo se incrementa con reiniciación una vez cada 5 pulsos de AdvanceLine. El orden de las líneas que se están escribiendo es simplemente 0-1-2-3-4-5-0-1-2-3 ... etc. De ese modo, las escrituras (25 x 1.280 x 3) se equilibran con las lecturas (19.200 x 5).

10.2.7 Memoria Intermedia 2 115

La Memoria Intermedia 2 acepta la salida procedente del proceso 112 de Resimple-CreateLuminance, donde se genera un píxel RGB y L completo para una coordenada de píxel dada. La salida de la Memoria Intermedia 2 115 va hasta el proceso 65 de Intensificación, lo que requiere un conjunto de 3 x 3 de valores 135 de luminancia centrados en el píxel que está siendo intensificado.

Por consiguiente, durante el proceso 66 de intensificación, existe la necesidad de un acceso a la matriz 3 x 3 de valores de luminancia, así como también al valor 136 RGB correspondiente para el píxel central de luminancia. Al mismo tiempo, los siguientes 3 valores de luminancia y el valor central RGB correspondiente deben ser calculados mediante el proceso 112 de Resimple-CreateLuminance. La vista lógica de los accesos a la Memoria Intermedia 2 115 se muestra en la Figura 49.

La implementación real de la Memoria Intermedia 2 115 se realiza simplemente como una RAM 4 x 6 (24 entradas) de 8 bits, proporcionando el direccionamiento sobre lectura y escritura el desplazamiento efectivo de valores. Un contador de columna de 2 bits, puede ser incrementado con el reinicio para proporcionar una memoria intermedia cíclica, que implementa de forma efectiva el equivalente de desplazamiento de los datos de la memoria intermedia completa en 1 posición de columna. El hecho de que no tengamos necesidad de la cuarta columna de datos RGB, no es relevante, y simplemente utiliza 3 bytes y el ahorro de no tener que implementar ningún desplazamiento ni lógica de lectura/ escritura complicados. En un ciclo dado, la RAM puede, o bien ser escrita o bien ser leída. Los procesos de lectura y de escritura tienen 75 ciclos para completar con el fin de seguir el ritmo de la cabeza de impresión.

10.2.8 Intensificación

La unidad 65 de Intensificación realiza la tarea de intensificación que se ha descrito en la Sección 3.2.6. Puesto que los píxeles RGB intensificados son almacenados en la Memoria Intermedia 3 116, la Unidad 65 de Intensificación debe seguir el ritmo del proceso 66 de Convertir en CMY, lo que implica que un píxel RGB completo debe ser intensificado dentro de los 745 ciclos.

El proceso de intensificación incluye un filtro pasa-alto de L (un canal generado a partir de los datos RGB y almacenado en la Memoria Intermedia 2), y añadir el L filtrado de nuevo a las componentes RGB, según se describió en la Tabla 12 dentro de la Sección 3.2.6.2. El filtro pasa-alto utilizado es un filtro pasa-alto básico que utiliza un función de convolución de 3 x 3, como se muestra en la Figura 50.

El filtro pasa-alto se calcula sobre 10 ciclos. El primer ciclo carga el registro 140 temporal con 8 veces el valor de píxel central (el píxel central desplazado a la izquierda en 3 bits). Los siguientes 8 ciclos restan los 8 valores de píxel restantes, con un fondo de 0. De ese modo, el procedimiento completo puede ser realizado mediante un sumador. El ciclo 10 incluye la multiplicación del resultado por una constante 141. Esta constante es la representación de 1/9, pero es un registro que permite que la cantidad sea alterada mediante software, mediante algún factor escalar.

La cantidad total se añade a continuación a los valores R, G y B (con un techo de 255), y se escriben en la Memoria Intermedia 3 durante los ciclos 72, 73 y 74. Calcular/ escribir los valores RGB intensificados durante los 3 últimos ciclos del conjunto de 75 ciclos, elimina la necesidad de un doble almacenamiento temporal intermedio en la Memoria Intermedia 3.

ES 2 282 136 T3

La estructura de la unidad de Intensificación puede ser apreciada en la Figura 51.

La unidad 142 de sumador conectada a la Memoria Intermedia 2 115, es un substractor con un fondo de 0. El TMP 140 se carga con 8 x el primer valor de L durante el ciclo 0 (de 75), y a continuación los siguientes 8 valores de L se restan del mismo. No se pone signo al resultado, puesto que la substracción tiene como fondo 0.

Durante el ciclo 10° (Ciclo 9), el total de 11 bits del TMP 140 se multiplica por un actor escalar (típicamente 1/9, pero bajo control de software, de modo que el factor puede ser ajustado), y se escribe de nuevo en el TMP 140. Solamente 8 bits enteros del resultado se escriben en el TMP (la fracción se desecha), de modo que el límite de unidad de multiplicar es 255. Si se utilizan un factor escalar de 1/9, el valor máximo escrito será 226 (255 x 8/9). El factor escalar es 8 bits de fracción, con el bit representando 1/8. El factor escalar variable puede tener en cuenta el hecho de que diferentes formatos de impresión son el resultado de escalar la imagen CFA con una cantidad diferente (y de ese modo, la convolución de 3 x 3 producirá resultados correspondientemente escalados).

Los valores intensificados para el rojo, el verde y el azul, se calculan durante el Ciclo 72, el Ciclo 73 y el Ciclo 74, y se escriben en los registros R, G y B de la Memoria Intermedia 3 116, una escritura por ciclo. El cálculo realizado en estos 3 ciclos consiste simplemente en la adición de TMP al R, G y B de la Memoria Intermedia 2 correspondiente al píxel central.

La Generación de Dirección es directa. Escribir en la Memoria Intermedia 3 116 es simplemente R, G y B en los ciclos 72, 73 y 74, respectivamente. La lectura en la memoria Intermedia 2 115 hace uso de la naturaleza cíclica de la Memoria 2 Intermedia. La dirección consiste en una componente de columna de 2 bits (que representa cuál de las 4 columnas debe ser leída), y un valor de 3 bits que representa L1, L2, L3, R, G o B. El número de columna empieza por 1 en cada línea, y se incrementa (con reinicio cíclico) cada 75 ciclos. El orden de lectura en la Memoria Intermedia 2 se muestra en la Tabla 28. El registro C es una componente de columna de 2 bits de la dirección. Toda adición sobre C es de módulo 4 (se reinicia cíclicamente dentro de 2 bits).

TABLA 28

Acceso de Lectura a la Memoria Intermedia 2 durante el conjunto de 75 Ciclos

Ciclo	Dirección	Actualizar C
0	C, L2	C = C-1
1	C, L1	
2	C, L2	
3	C, L3	C = C+1
4	C, L1	
5	C, L3	C = C+1
6	C, L1	
7	C, L2	
8	C, L2	C = C-1
9-71	Ningún acceso	
72	C, R	
73	C, G	
74	C, B	C = C-1

Tras el Ciclo 74, el registro C conserva el número de columna para el siguiente conjunto de cálculo, haciendo así que la búsqueda durante el siguiente Ciclo 0 sea válida.

La intensificación puede empezar solamente cuando existan suficientes píxeles L y RGB escritos en la Memoria Intermedia 2 (de modo que el filtro pasa-alto sea válido). El proceso de intensificación debe detenerse por lo tanto hasta que el proceso de escritura de la Memoria Intermedia 2 haya avanzado 3 columnas.

ES 2 282 136 T3

10.2.9 Memoria Intermedia 1 114

La Memoria Intermedia 1 mantiene los píxeles de blanco balanceado y rango expandido a la resolución espacial de captura original. Cada píxel se almacena con 10 bits de resolución de color, en comparación con la resolución de 8 bits por píxel de resolución de color de almacenamiento de imagen de la RAM de imagen.

La Memoria Intermedia 1 está dispuesta como 3 memorias intermedias direccionables por separado: una para cada plano de color del rojo 145, verde 146 y azul 147. Se muestra un compendio simple de las memorias intermedias en la Figura 52.

Durante el transcurso de los 75 ciclos, se leen 16 entradas desde cada una de las 3 memorias intermedias 3 veces por medio del proceso 112 de Remuestreo, y se escriben hasta 29 nuevos valores en las 3 memorias intermedias (el número exacto depende del factor escalar y de la posición actual de sub-píxel durante el remuestreo).

Las memorias intermedias deben ser suficientemente amplias como para la lectura y la escritura se pueda realizar sin interferir una con otra. Durante el proceso de lectura, se leen 4 píxeles desde cada una de las 6 filas. Si el factor escalar es muy grande (por ejemplo, escalamos a Panoramic), los mismos píxeles de entrada pueden ser leídos múltiples veces (utilizando una posición de función diferente para el remuestreo). Eventualmente, sin embargo, se requerirán los siguientes píxeles. Si no escalamos tanto, los nuevos píxeles pueden ser requeridos antes de 1 siguiente ciclo de generación de píxel (es decir, dentro de los 75 ciclos de reloj).

Observando los factores escalares de la Tabla 9 y de la Tabla 11, el peor caso para el escalado es el formato Passport:

- El plano verde tiene un valor Δ para Passport de 1,5625, lo que indica que 4 posiciones pueden estar contenidas dentro de 6 posiciones de píxel CFA. Sin embargo, cada fila de muestras verdes solamente mantiene cada píxel alterno. Esto significa que solamente se necesitan 4 muestras por fila (el peor caso es 4, no 3, debido al peor caso de posición inicial). El movimiento en Y indica el requisito de una columna de muestra adicional, que hace 5. Finalmente, se requiere una columna de muestra adicional para la escritura. Esto da un total de 6 muestras por fila. Se requieren 7 filas para una muestra simple. Para generar los 3 conjuntos de píxeles RGB para cada posición x, el movimiento máximo en y será de 4 filas ($3,125 = 2 \times 1,5625$). El movimiento X añade una fila de muestra por arriba y por abajo. Por consiguiente, se requiere un total de 13 filas. Para más detalles, véase la Sección 10.2.10.
- Los planos rojo y azul tienen un valor Δ para Passport de 0,78125, lo que indica que 4 posiciones pueden estar contenidas en el interior de 4 muestras. Se requiere una muestra adicional para escribir mientras que las 4 restantes están siendo leídas. Esto da un total de 5 muestras por fila, que se incrementa además hasta 6 muestras para emparejarse con el plano verde (a efectos de arranque). Se requieren 6 filas para proporcionar movimiento en y. Para más detalles, véase la Sección 10.2.10.

Cada sub-memoria intermedia se implementa como una RAM con descodificación para leer o escribir una muestra simple de 10 bits por ciclo. Las sub-memorias intermedias se han resumido en la Tabla 29, y consumen menos de 200 bytes.

TABLA 29

Sumario de Sub-Memoria Intermedia

Memoria Intermedia	Composición	Bits
Memoria Intermedia del Rojo	6 filas x 6 muestras x 10 bits	360
Memoria Intermedia del Azul	6 filas x 6 muestras x 10 bits	360
Memoria Intermedia del Verde	12 filas x 6 muestras x 10 bits	780
	TOTAL	1.500

10.2.10 Remuestrear y Crear Canal de Luminancia

El proceso 112 de Remuestrear y Crear Canal de Luminancia es responsable de la generación del valor de píxel RGB en un espacio de resolución media mediante un remuestreo apropiado de las imágenes planares R, G y B de rango expandido y blanco balanceado, como se describe en la Sección 3.2.5. Adicionalmente, los valores de luminancia para el píxel RGB dado, así como también los valores de luminancia para el píxel por encima y por debajo del píxel RGB, deben ser generados para su uso en el último proceso de intensificación.

ES 2 282 136 T3

El tiempo permitido para producir el valor RGB y 3 valores L es de 75 ciclos. Dado que L es simplemente la media del mínimo y el máximo de R, G y B para una posición de píxel dada (véase la Sección 3.2.6.I), debemos producir efectivamente valores RGB para 3 coordenadas de píxel: el píxel en cuestión y el píxel de por encima y por debajo. Así, disponemos de 75 ciclos en los que calcular las 3 muestras RGB de resolución media y sus valores L correspondientes.

Introducir en memoria intermedia valores L (y por tanto valores RGB) para salvar el recálculo, requiere demasiada memoria, y en cualquier caso, tenemos tiempo suficiente para generar los valores RGB. La Memoria Intermedia 4 117 contiene píxeles de resolución media, pero no pueden ser utilizados debido a que contiene píxeles CMY intensificados (en vez de píxeles RGB no intensificados).

10.2.10.1 Remuestreo

El proceso de remuestreo puede ser visto como 3 conjuntos de generación RGB, cada uno de los cuales debe ser completado dentro de 25 ciclos (para un tiempo máximo total transcurrido de 75 ciclos). El proceso de generación de un valor RGB simple puede ser visto, a su vez, como 3 procesos realizados en paralelo: el cálculo de R, el cálculo de G, y el cálculo de B, todo ello para una coordenada de píxel de resolución media dada. La teoría para generar cada uno de estos valores, puede ser encontrada en la Sección 3.2.5, pero la conclusión consiste en ejecutar efectivamente tres filtros de reconstrucción de imagen, uno en cada canal de la imagen. En el caso del PCP, realizamos la reconstrucción de imagen con 5 puntos de muestro, requiriendo 4 coeficientes en la función de convolución (puesto que un coeficiente es siempre 0 y de ese modo no se requiere el punto de muestra).

Por consiguiente, el cálculo del píxel R de resolución media se consigue ejecutando un filtro de reconstrucción de imagen sobre los datos R. El cálculo del píxel G de resolución media se logra ejecutando un filtro de resolución de imagen sobre los datos G, y el cálculo del píxel B de resolución media se logra ejecutando un filtro de reconstrucción de imagen sobre los datos B. Aunque las funciones son simétricas en x e y, no son iguales para cada plano de color. R y B son probablemente la misma función debido a sus características similares de imagen, pero el plano G, debido a la rotación requerida para la reconstrucción de imagen, debe tener una función diferente. La vista de alto nivel del proceso puede ser apreciada en la Figura 53. La generación de dirección no ha sido mostrada.

El proceso de remuestreo puede comenzar solamente cuando existen píxeles suficientes en la Memoria Intermedia 1 para la línea de píxel actual que se está generando. Este será el caso una vez que se hayan escrito 4 columnas de datos en cada uno de los planos de color en la Memoria Intermedia 1 114. El proceso 112 de remuestreo debe detenerse hasta ese momento.

Para calcular un valor de píxel de resolución media de un plano de color dado, tenemos de 25 ciclos disponibles. Para aplicar la función área de muestra 4×4 , aplicamos la función 1D (indexada mediante x) sobre cada una de las 4 filas de 4 muestras de entrada. A continuación aplicamos la función 1D (indexada mediante y) sobre los 4 valores de píxel resultantes. El resultado final es el píxel de salida remuestreado. La aplicación de un único coeficiente a cada ciclo proporciona un total de 16 ciclos para generar los r valores intermedios, y 4 ciclos para generar el valor de píxel final, para un total de 20 ciclos.

Con respecto a la precisión, los píxeles de entrada son de 10 bits cada uno de ellos (8:2), y los coeficientes de la función son de 12 bits. Mantenemos 14 bits de precisión durante las 4 etapas de cada aplicación de la función (8:6), pero solamente salvamos 10 bits para el resultado (8:2). Así, se puede utilizar el mismo ingenio de convolución cuando la convolución se realice en x e y. La salida final de R, G o B es de 8 bits.

El corazón del proceso de remuestreo lo constituye la Unidad de Convolución 50, como se muestra en la Figura 54.

El proceso de remuestreo incluye entonces 20 ciclos, como se muestra en la Tabla 30. Obsérvese que la Fila 1, Píxel 1, etc., se refiere a la entrada desde la Memoria Intermedia 1 114, y debe ser tomado en consideración por el mecanismo de direccionamiento (véase más abajo).

ES 2 282 136 T3

TABLA 30

Remuestreo de 20 Ciclos

5

10

15

20

25

30

35

Ciclos	Función	Aplicar Función a:	Almacenar Resultado en
1	X[1]	Fila 1, Píxel 1	TMP
2	X[2]	Fila 1, Píxel 2	TMP
3	X[3]	Fila 1, Píxel 3	TMP
4	X[4]	Fila 1, Píxel 4	TMP, V1
5	X[1]	Fila 2, Píxel 1	TMP
6	X[2]	Fila 2, Píxel 2	TMP
7	X[3]	Fila 2, Píxel 3	TMP
8	X[4]	Fila 2, Píxel 4	TMP, V2
9	X[1]	Fila 3, Píxel 1	TMP
10	X[2]	Fila 3, Píxel 2	TMP
11	X[3]	Fila 3, Píxel 3	TMP
12	X[4]	Fila 3, Píxel 4	TMP, V3
13	X[1]	Fila 4, Píxel 1	TMP
14	X[2]	Fila 4, Píxel 2	TMP
15	X[3]	Fila 4, Píxel 3	TMP
16	X[4]	Fila 4, Píxel 4	TMP, V4
17	X[1]	V1	TMP
18	X[2]	V2	TMP
19	X[3]	V3	TMP
20	X[4]	V4	TMP (para salida)

10.2.10.2 Generación de L 8-

40

Según se ha descrito en la Sección 3.2.6.1, debemos convertir 80 desde RGB a L para el posterior proceso de intensificación. Consideramos el espacio de color L*a*b CIE 1976, donde L es perceptivamente uniforme. Para la conversión desde RGB a L (el canal de luminancia), promediamos el mínimo y el máximo de R, G y B, como sigue:

45

$$L = \frac{\text{MÍN}(R, G, B) + \text{MÁX}(R, G, B)}{2}$$

50

La generación de valores R, G y B de un píxel dado, se realiza en paralelo, tomando 20 ciclos. El tiempo total para la generación de L según se describe aquí, es de 4 ciclos. Esto hace que el tiempo total de generación de un conjunto de píxel RGBL sea de 24 ciclos, con 1 ciclo para compartir (puesto que el proceso debe ser completado dentro de los 25 ciclos).

55

El valor de L puede ser así escrito de manera segura en la Memoria Intermedia 2 115 en el ciclo 25°. La generación de dirección se describe a continuación.

60

Un comparador simple de 8 bits puede producir 3 bits en 3 ciclos, que pueden ser utilizados posteriormente para seleccionar las 2 entradas para el sumador, como se muestra en la Tabla 31. La división por 2 puede ser incorporada de simplemente en el sumador.

65

ES 2 282 136 T3

TABLA 31

Selección de Mín y Máx en base a 3 comparaciones

MÍN	MÁX	R>G	G>B	R>B
R	B	1	1	X ^a
R	G	1	0	1
G	R	0	1	0
G	B	0	1	1
B	R	0	0	X
B	G	1	0	0

a. Sin importar el estado

Puesto que el sumador simplemente añade el valor mínimo al máximo, el orden no es importante. Por consiguiente, de las 2 entradas al sumador, Input1 puede ser una opción entre R y G, mientras que Input2 es una opción entre G y B. La lógica es una minimización de los patrones de bit apropiados de la Tabla 31.

10.2.10.3 Generación de Dirección para la Memoria Intermedia 2

La salida desde el Remuestreador es un píxel RGB simple, y 3 píxeles de luminancia (L) centrados verticalmente sobre el píxel RGB. Los 3 valores de L pueden ser escritos en la Memoria Intermedia 2, uno cada 25 ciclos. Los valores R, G y B deben ser escritos después del ciclo 45 y antes del ciclo 50, puesto que el segundo píxel generado es el píxel central cuyos valores RGB deben ser mantenidos. La dirección Buffer2 consiste en un componente de columna de 2 bits (que representa cuál de las 4 columnas ha de ser escrita), y un valor de 3 bits que representa L1, L2, L3, R, G, o B. El número de columna empieza en 0 en cada línea, y se incrementa (con reinicio cíclico) cada 75 ciclos (es decir, después de escribir L3).

10.2.10.4 Generación de Dirección para Búsqueda de Función

El método de cálculo de la dirección de la función es el mismo que se ha descrito al final de la Sección 3.2.5. Cada función es de 1 dimensión, con 64 entradas en la tabla. Los 6 bits más significativos (truncados) de componente fraccional en el espacio de función actual, se utilizan para indexar en la tabla de coeficientes de función. Para los 16 primeros ciclos, la ordenada X se utiliza para indexar la función, mientras que en los siguientes 4 ciclos se utiliza la ordenada Y. Puesto que la función es simétrica, se puede utilizar la misma función para ambas X e Y.

Para cada uno de los 1.280 valores remuestreados, necesitamos producir 3 píxeles: el píxel en cuestión 161, y los píxeles de por encima 160 y por debajo 162 de ese píxel. En vez de generar un píxel central y después moverse arriba y abajo desde el píxel central, generamos un píxel 160 y generamos dos píxeles 161, 162 por debajo del mismo. El segundo píxel 161 generado se toma el píxel central. A continuación volvemos a la fila original y generamos los siguientes 3 píxeles en la siguiente posición de salida. De esta manera, como muestra la Figura 55, generamos 3 píxeles para cada una de las 1.280 posiciones.

De ese modo, disponemos de una posición actual en el espacio de la función. Según avanzamos hasta el siguiente píxel en X o Y en el espacio de entrada original, añadimos valores de datos apropiados a estas coordenadas de función. Observando la Figura 56, vemos dos casos de espacio de entrada girado y no girado.

Consideremos que el movimiento en X e Y como ΔX y ΔY , siendo sus valores dependientes del formato de impresión, y por tanto del valor de mps (véase la Sección 3.2.5). Para el canal del verde, $\Delta X = \Delta Y = \frac{1}{2}$ mps. Para los canales del rojo y del azul, $\Delta X = 1$ /mps y $\Delta Y = 0$. Véase la Tabla 9 y la Tabla 11 para valores apropiados de ΔX y de ΔY .

Ahora podemos aplicar los valores ΔX y ΔY al movimiento dentro de la función. Por consiguiente, cuando avanzamos en X, añadimos ΔX a X y restamos ΔY de Y. En el caso no girado, simplemente se resta 0 de Y. De igual modo, cuando avanzados en Y, añadimos ΔY a X y ΔX a Y. Podemos hacer esto debido a que el movimiento en X y en Y difiere en 90 grados.

La generación de dirección para la búsqueda de función supone una posición de partida establecida mediante software, y dos deltas ΔX y ΔY con respecto al movimiento en Y en el espacio de la función. La lógica de generación de función se muestra en pseudo-código que sigue:

```

ColumnKernelY = StartKernelY
ColumnKernelX = StartKernelX
5  Hacer NLines veces (sin embargo, existen muchas líneas de salida por procesar)
    KernelX = ColumnKernelX
    KernelY = ColumnKernelY
10  Hacer 1.280 veces
    GeneratePixel
    KernelX = KernelX + DeltaY (movimiento en Y)
15  KernelY = KernelY + DeltaX (movimiento en Y)
    GeneratePixel
    KernelX = KernelX + DeltaY (movimiento en Y)
20  KernelY = KernelY + DeltaX (movimiento en Y)
    GeneratePixel
25  KernelX = ColumnKernelX + Deltas (movimiento en X)
    KernelY = ColumnKernelY + DeltaY (movimiento en X)
    EndDo
30  ColumnKernelY = ColumnKernelY + DeltaX (movimiento en Y)
    ColumnKernelX = ColumnKernelX + DeltaY (movimiento en Y)
35  EndDo

```

Según se muestra en el pseudo-código, la generación de 3 píxeles se produce 1.280 veces. Asociadas a la generación de cada píxel, están 2 adiciones, las cuales pueden ser realizadas durante el transcurso de la tarea de 25 ciclos GeneratePixel. Cada tarea GeneratePixel es de 25 ciclos, consistiendo en 4 conjuntos de 4 ciclos que indexan la función por medio de KernelX (coeficientes 0, 1, 2, 3), seguidos de 4 ciclos que indexan la función a través de KernelY (coeficientes 0, 1, 2, 3), seguidos de 9 ciclos de espera.

Obsérvese que todos los valores son positivos y solamente fraccionales. Las dos realizaciones a partir de la actualización de los valores X e Y de la función, se disponen a la salida para la generación de dirección de la Memoria Intermedia 1 (véase la Sección 10.2.10.5). Estas banderolas de realización indican simplemente si las ordenadas particulares para la función se reiniciaron cíclicamente o no durante la operación matemática. El reinicio cíclico puede ser o bien por encima de 1 o bien por debajo de 0, pero el resultado es siempre positivo.

Los dos bits de realización son enviados siempre a la Unidad de Girar/ Balance del Blanco/ Expansión de Rango, para su uso en la determinación de líneas de entrada relativas de la imagen.

10.2.10.5 Generación de Dirección para la Memoria Intermedia 1

El Remuestreador 112 lee desde la Memoria Intermedia 1 114, que consiste en 3 memorias intermedias 145, 146, 147 direccionables individualmente: una por cada plano de color. Cada memoria intermedia puede ser, o bien leída o bien escrita, durante cada ciclo.

El proceso de lectura de 75 ciclos se fracciona en 3 conjuntos de 25 ciclos: un conjunto de 25 ciclos para la generación de cada píxel. Cada conjunto de 25 ciclos incluye 16 lecturas de la Memoria Intermedia 1, seguidas de 9 ciclos sin ningún acceso. La Memoria Intermedia 1 se escribe durante estos 9 ciclos. Las 16 lecturas de la Memoria Intermedia 1 114, son efectivamente 4 conjuntos de 4 lecturas, y coinciden con 4 grupos de 4 lecturas en la función para cada plano de color.

La generación de dirección incluye entonces la generación de 16 direcciones para calcular el primer píxel (seguido de 9 ciclos de espera), la generación de 16 direcciones para calcular el segundo píxel (seguido de 9 ciclos de espera), y finalmente la generación de las 16 direcciones para el tercer píxel (seguido de 9 ciclos de espera).

Cada plano de color tiene sus propios parámetros de dirección de Memoria Intermedia 1 de partida. Según son generados los 3 conjuntos de direcciones para cada una de las 1.280 posiciones a lo largo de la línea, y según avanza el muestreador desde una línea de 1.280 muestras hasta la siguiente, los dos bits de realización procedentes de la Unidad de Generación de Dirección de Función son utilizados para actualizar estos parámetros de dirección de la Memoria Intermedia 1.

10.2.10.6 Memoria Intermedia del Verde 146

La generación de dirección para la sub-memoria intermedia 146 del verde en el interior de la Memoria Intermedia 1 114, es más complicada que en la sub-memoria intermedia 145 del rojo y en la sub-memoria intermedia 147 del azul, por dos razones principales:

- el canal del verde representa un patrón a modo de tablero de ajedrez en el CFA. Las líneas alterna consisten en píxeles impares o pares solamente. Para remuestrear el canal del verde, debemos girar efectivamente el canal 45 grados.
- existe el doble de píxeles verdes que de píxeles rojos o azules. Remuestrear significa la lectura de más muestras en la misma cantidad de tiempo: existen aún 16 muestras leídas para generar cada píxel en el espacio de resolución media, pero existe una probabilidad más alta de hacer avanzar la memoria intermedia cada vez. La probabilidad exacta depende del factor escalar utilizado.

Sin embargo, se utiliza el mismo concepto de utilización de una RAM como memoria intermedia cíclica para el canal del verde. La sub-memoria intermedia del verde es una RAM de 78 entradas con una estructura lógica de 13 filas, conteniendo cada una de ellas 6 entradas. La relación entre la dirección de RAM y la posición lógica se muestra en la Figura 57.

Las muestras en la Memoria Intermedia 1 146 representan un patrón a modo de tablero de ajedrez en el CFA. En consecuencia, las muestras de una fila (por ejemplo, las direcciones 0, 13, 26, 39, 52, 65) pueden representar píxeles impares o pares, dependiendo de la línea actual dentro de la imagen completa, y si la imagen ha sido girada o no en 90 grados. Esto se ilustra en la Figura 58.

Por consiguiente, cuando mapeamos un área de muestra de 4 x 4 en la memoria intermedia, existen dos posibilidades para la interpretación de las muestras. Como resultado, existen dos tipos de direccionamiento, dependiendo de si la línea actual está representada por píxeles impares o pares. Esto significa que las filas pares con rotación de imagen 0 tienen el mismo direccionamiento que las filas impares con rotación de imagen 90, puesto que ambas contienen píxeles impares. De igual modo, las filas impares con rotación de imagen 0 tendrán el mismo direccionamiento que las filas pares con rotación de imagen 90, puesto que ambas contienen píxeles pares. La decisión se ha resumido en la Tabla 32.

TABLA 32

Determinación del Tipo de Muestreo

Rotación	Línea Actual		píxeles	Tipo
0	Línea Par	8	Impar	Tipo 2
0	Línea Impar	8	Par	Tipo 1
90	Línea Par	8	Par	Tipo 1
90	Línea Impar	8	Impar	Tipo 2

La ventana de muestreo rea 4 x 4, es la manera en que giramos efectivamente la memoria intermedia en 45 grados. La rotación de 45 grados es necesaria para el remuestreo efectivo, como se describe en la Sección 3.2.5.

Suponiendo por el momento que solamente necesitamos generar un remuestreo simple, consideramos el direccionamiento de la memoria intermedia mediante el examen de dos tipos de ventanas de muestreo 4 x 4 como se muestra en la Figura 59.

Aunque los dos tipos de muestreo 4 x 4 parecen similares, la diferencia se deriva de la forma en que se representa el mapeo 4 x 4 en la imagen planar. La Figura 60 ilustra el mapeo del muestreo 4 x 4 de Tipo 1 en la sub-memoria intermedia del verde. Solamente las 7 filas superiores y las 4 columnas más a la derecha han sido representadas, puesto que el área de la muestra 4 x 4 está contenida completamente dentro de esta área.

ES 2 282 136 T3

El mapeo de píxeles de memoria intermedia para muestrear las filas para el proceso de muestreo de Tipo 2, es muy similar y puede ser apreciado en la Figura 61.

En ambos Tipo 1 y Tipo 2 de direccionamiento de las 16 muestras existen dos formas de procesamiento de una fila. El procesamiento del direccionamiento de las Filas 1 y 3 de Tipo 1 es el mismo (relativamente hablando) que el procesamiento de las filas 2 y 3 de Tipo 2. De igual modo, el procesamiento de las filas 2 y 4 de Tipo 1 es el mismo (relativamente hablando) que el procesamiento de las filas 1 y 3 de Tipo 2. Denominaremos a estos métodos de direccionamiento de fila Tipo A 170 y Tipo B 171, como se muestra en la Figura 62.

Dada una posición de partida para la ventana 4 x 4 (WindowStartAdr) y un tipo de partida (WindowStartType), podemos generar las direcciones para las 16 muestras por medio de una tabla de 8 entradas (para atravesar los dos conjuntos de 4 muestras). Cuando leemos el primer valor de muestra, añadimos una desviación de la tabla para llegar a la siguiente posición. La desviación dependerá del tipo (A, B = 0, 1). La desviación desde la cuarta muestra es la cantidad necesaria para llegar al primer punto de muestra para la siguiente línea (y debe tener en cuenta el número de columnas de muestra). Tras generar cada fila de 4 muestras, intercambiamos Tipo A y Tipo B. La lógica para generar las direcciones para un conjunto simple de 16 muestras, aparece en el pseudo-código que sigue. El módulo 78 de adición proporciona el almacenamiento cíclico.

```

20  _____
    Adr = WindowStartAdr
    TypeAB = WindowStartType
    Hacerlo 4 veces
25      Para N = 0 a 4
          Ir a Adr
          Adr = (Adr + Table[TypeAB, N]) mod 78
30      EndFor
          TypeAB = NO TypeAB
35  EndDo
    _____
  
```

La tabla de búsqueda consiste en 8 entradas: 4 para la generación de desviación de dirección de Tipo A 170, y 4 para la de Tipo B 171. Las desviaciones son todas relativas a la posición de muestra actual (Adr).

TABLA 33

Valores de Desviación para la Generación de Dirección de 16 Muestras

TypeAB	N	Desviación
0	0	14
0	1	1
0	2	14
0	3	37
1	0	1
1	1	14
1	2	1
1	3	37

Al final de las 16 lecturas, el bit TypeAB será el mismo que el valor original (cargado desde WindowStartType).

Leer un solo conjunto de 16 muestras no es suficiente. Se deben leer tres conjuntos de 16 muestras (que representan 3 posiciones diferentes en Y en el espacio de entrada no girado). Al final del primer y segundo conjuntos de 16 muestras, las posiciones de la función son actualizadas mediante el generador de dirección de función. Los bits de

ES 2 282 136 T3

realización de esta actualización son utilizados para establecer la ventana para el siguiente conjunto de 16 muestras. Los dos bits de realización se indexan en la tabla que contiene una desviación y una banderola de 1 bit. La desviación se añade a WindowStartAdr, y la banderola se utiliza para determinar si se invierte o no WindowStartType. Los valores para la tabla se muestran en la Tabla 34.

TABLA 34

Actualización de WindowStartAdr y WindowStartType

Realización de KernelX	Realización de KernelY	Desviación	Tipo
0	0	0	Sin cambio
0	1	1	Invertir
1	0	14	Invertir
1	1	2	Sin cambio

Al final del tercer conjunto de 16 muestras, las posiciones de la función son actualizadas para compensar en avance en X en el espacio de entrada no girado. Esta vez, se produce una dirección de movimiento diferente, por lo que se utiliza una tabla de modificación de Offset/TypeAB diferente. No podemos añadir estas desviaciones al valor WindowStartAdr actual, debido a que representa una posición de dos movimientos en Y hacia fuera de donde deseamos iniciar el movimiento. En consecuencia, cargamos WindowStartAdr y WindowStartType desde otro conjunto de variables: TopStartAdr y TopStartType, que representan la primera entrada en la línea actual de 1.280. Las dos banderolas de realización procedentes del generador de dirección de la Función, se utilizan para la búsqueda en la Tabla 35, para determinar la desviación que se ha de añadir a TopStartAdr y si se invierte o no TopStartType. Como antes, la adición es el módulo 78 (el tamaño de la RAM del verde). Los resultados se copian en WindowStartAdr y WindowStartType par su uso en la generación de los siguientes 3 conjuntos de 16 muestras.

TABLA 35

Actualización de TopStartArt y TopStartType

Realización de KernelX	Realización de KernelY	Desviación	Tipo
0	0	0	Sin cambio
0	1	12	Invertir
1	0	14	Invertir
1	1	13	Sin cambio

Tras el procesamiento de los 1.280 conjuntos de 3 conjuntos de 16 muestras, empieza la siguiente línea de 1.280. Sin embargo, la dirección de la primera muestra para la posición 0 dentro de la siguiente línea debe ser determinada. Puesto que las muestras son siempre cargadas en los lugares correctos en la Memoria Intermedia 1, podemos empezar siempre exactamente desde la misma posición en la Memoria Intermedia 1 (es decir, TopStartAdr puede ser cargado de una Position0Adr constante). Sin embargo, debemos preocuparnos sobre qué tipo estamos tratando, puesto que el tipo depende de cuánto avanzamos. Por consiguiente, tenemos un Position0Type inicial que debe ser actualizado dependiendo de las banderolas de realización desde el generador de dirección de función. Puesto que nos estamos moviendo en el espacio de entrada Y no girado, la lógica utilizada es la misma que para la actualización del WindowStartType, salvo en que se realiza sobre Position0Type. El nuevo valor para Position0Type se copia en TopStartType, y en WindowStartAdr para empezar el muestreo de la primera posición de la nueva línea.

El proceso de muestreo para una línea de posición de 1.280 dada, no puede empezar hasta que existan entradas suficientes en al Memoria Intermedia 1, situadas ahí por la Unidad de Giro/ Balance del Blanco/ Expansión de Rango. Esto ocurrirá 128 ciclos después de cada nueva línea (véase la Sección 10.2.11).

10.2.10.7 Memorias intermedias del Rojo y del Azul

La sub-memoria intermedia 145 del rojo y la sub-memoria intermedia 147 del azul de la Memoria Intermedia 1, son simplemente 2 RAMs a las que se accede como memorias intermedias cíclicas. Cada memoria intermedia es de

ES 2 282 136 T3

30 bytes, pero tiene una organización lógica de 6 filas, cada una de las cuales contiene 6 entradas. La relación entre la dirección de RAM y la posición lógica, se muestra en la Figura 63.

5 Para el rojo y el azul, las primeras 16 muestras que han de ser leídas son siempre las 4 x 4 entradas superiores. No se puede tener acceso a las dos columnas de muestras restantes por parte del algoritmo de lectura en esta fase.

Las generación de dirección para estas primeras 16 muestras es simplemente una posición de inicio (en este caso 0) seguido de 16 etapas de módulo 36 de adición, como se muestra en el siguiente pseudo-código:

10

```

ADR = StartADR
Hacerlo 4 veces
    Hacerlo 4 veces
        ADR = ADR + 6 MOD 36
    End Do
ADR = ADR + 13 MOD 36
End Do
    
```

15

20

25

30 Sin embargo, este mecanismo de generación de dirección es diferente del canal del verde. En vez de diseñar dos mecanismo de dirección, es posible aplicar el esquema de direccionamiento del verde a los canales de rojo y de azul, y simplemente utilizar valores diferentes en las tablas. Esto reduce la complejidad del diseño. La única diferencia entonces, reside en el módulo 36 de adición, en vez del módulo 78 de adición. Esto puede ser proporcionado por un simple multiplexor.

35

Viendo las diversas tablas de dirección para el verde, y considerándolas como aplicadas al rojo y al azul, es evidente que no existe ninguna necesidad de un Tipo, puesto que ambos canales del rojo y del azul no necesitan ser girados 45 grados. De ese modo, podemos ignorar de forma segura el valor Tipo. El equivalente rojo/ azul de la Tabla 33, mostrado en la Tabla 36, posee dos conjuntos de 4 entradas idénticas.

40

TABLA 36

Valores de Desviación para Generación de Dirección de 16 Muestras (Rojo / Azul)

45

TypeAB	N	Desviación
0	0	6
0	1	6
0	2	6
0	3	13
1	0	6
1	1	6
1	2	6
1	3	13

50

55

60 Al igual que con la generación de la dirección del verde, nos movemos dos veces en Y antes de avanzar hasta la siguiente entrada de 1.280. Para el rojo y el azul no existe ningún escalado entre el movimiento en el espacio de la función y el movimiento den el espacio de entrada. No existe tampoco ninguna rotación. Según nos movemos en Y, el ΔY de 0 se añade a KernelX (véase generación de dirección de función en la Sección 10.2.10.4). Como resultado, nunca se establecerá la realización de KernelX. Mirando la Tabla 34, las únicas ocurrencias posibles son los valores 00 ó 01 KernelX/ KernelY. En el caso de 00, la solución del verde es no cambiar a ninguno de WindowStartAdr ni WindowStartType, de modo que esto es también correcto para el rojo y el azul. En el caso de 01, deseamos añadir 1 a WindowStartAdr, y no hay que tener ningún cuidado acerca de WindowStartType. Los valores del verde pueden ser utilizados por lo tanto de manera segura para el rojo y el azul. El peor caso es el avance en 1 en la dirección ambas veces, dando como resultado un caso peor de solapamiento como se muestra en la Figura 65.

65

ES 2 282 136 T3

Al final del tercer conjunto de 16 muestras, TopStartAdr y TopStartType deben ser actualizados. Puesto que nos estamos moviendo en X (y añadiendo $\Delta Y = 0$ a KernelY), la realización desde KernelY será siempre 0. El equivalente del rojo/ azul de la Tabla 35 se muestra aquí en la Tabla 37. Obsérvese que no existe ninguna columna de Tipo, puesto que Tipo no es importante para el Rojo o el Azul.

TABLA 37

Actualización de TopStartAdr y de TopStartType (Rojo/ Azul)

Realización de KernelX	Realización de KernelY	Desviación
0	0	0
0	1	-
1	0	6
1	1	-

El proceso de avance desde una línea de 1.280 conjuntos de 3 píxeles hasta la siguiente, es el mismo que para el verde. La Position0Adr será la misma para el primer conjunto de 16 muestras para una línea dada (Position0Adr = 0 para el rojo y el azul), y Tipo es irrelevante. La generación del rojo y del azul debe iniciarse al mismo tiempo que la generación del verde, de modo que no puede empezar hasta 128 ciclos después del comienzo de una nueva línea (véase la Sección 10.2.11).

10.2.11 Giro, Balance del Blanco y Expansión de Rango 111

La tarea real de cargar la Memoria Intermedia 1 114 desde la RAM de imagen 111, incluye las etapas de rotación, balance del blanco, y expansión de rango 111, como se describió en la Sección 3.2.3 y en la Sección 3.2.4. Los píxeles deben ser producidos para la Memoria Intermedia 1 lo suficientemente rápidos para su uso por parte del proceso 112 de Remuestreo. Esto significa que durante un grupo simple de 75 ciclos, esta unidad debe estar capacitada para leer, procesar, y almacenar 6 píxeles rojos, 6 píxeles azules y 13 píxeles verdes.

La etapa de rotación opcional se acomete con la lectura de los píxeles en el orden apropiado. Una vez que un píxel dado ha sido leído desde el plano apropiado en el almacenamiento de la imagen, debe ser balanceado en el blanco y su valor ajustado de acuerdo con el cálculo de expansión de rango definido en la Sección 3.2.4. El proceso incluye simplemente una única substracción (fondo 0), y una multiplicación (techo 255), ambas frente a constantes específicas del color. La estructura de esta unidad se muestra en la Figura 66.

Los umbrales 72 bajos del rojo, verde y azul, junto con los factores 173 escalares del rojo, verde y azul, se determinan mediante la CPU 10 tras la generación del histograma para cada plano de color por medio de la Unidad 8 de Histograma de Imagen (véase la Sección 9).

Dependiendo de si el píxel actual que está siendo procesado en la línea es rojo, verde o azul, se multiplexan el umbral bajo y el factor escalar apropiados en la unidad de substracción y en la unidad de multiplicación, con la salida escrita en el plano de color apropiado en la Memoria Intermedia 1.

La unidad de Substracción 172 subtrae el valor de Umbral bajo de 8 bits desde el valor de píxel de RAM de Imagen de 8 bits, y tiene un fondo de 0. El resultado de 8 bits se hace pasar hasta la unidad de multiplicación especializada 8 x 8, la cual multiplica el valor de 8 bits por el factor escalar de 8 bits (8 bits de fracción, parte entera = 1). Solamente se mantienen los 10 bits superiores del resultado, y representan 8 bits de la parte entera y 2 bits de la fracción. El multiplicador 174 tiene un techo de resultado de 255, de modo que si algún bit mayor de 7 bits hubiera sido establecido como resultado de la multiplicación, el resultado de la parte entera completa de 8 bits se establece en 1s, y la parte fraccional se establece en 0.

Aparte de la unidad 172 de substracción y de la unidad 174 de multiplicación, la mayor parte del trabajo de esta unidad se realiza mediante el Generador de Dirección 175, el cual constituye efectivamente la máquina de estado para la unidad. La generación de dirección está gobernada por dos factores: en un ciclo dado, solamente se puede realizar un acceso a la RAM 11 de Imagen, y en un ciclo dado, solamente se puede realizar un acceso a la Memoria Intermedia 1 114. De los 75 ciclos disponibles, se utilizan 3 conjuntos de 16 ciclos para la lectura de la Memoria Intermedia 1. El uso real es de 3 conjuntos de 25 ciclos, con 16 lecturas seguidas de 9 ciclos de espera. Eso da un total de 27 ciclos disponibles para 25 escrituras 86 rojas, 6 azules, 6 verdes). Esto significa que se satisfacen las dos restricciones si la temporización de las escrituras en la Memoria Intermedia 1 coincide con los ciclos de espera del Remuestreador 112.

10.2.11.1 *Generación de Dirección para la Memoria Intermedia 1*

Una vez que se está ejecutando el proceso de remuestreo, solamente estamos afectados por la escritura en la Memoria Intermedia 1 durante el período en que el Remuestreador 112 no está leyendo en la misma. Puesto que el Remuestreador tiene 3 conjuntos de 16 lecturas cada período de 75 ciclos, existen 27 ciclos disponibles para la escritura. Cuando el remuestreador no está trabajando, deseamos cargar la Memoria Intermedia 1 tan rápido como sea posible, lo que significa una escritura en la Memoria Intermedia 1 114 cada ciclo. La Generación de Dirección para la Memoria Intermedia 1 se ejecuta fuera de una máquina de estado que toma en consideración estos dos casos. Siempre que se cargue un valor desde al RAM de Imagen 111, el valor ajustado se escribe en el color apropiado en la Memoria Intermedia 1 un ciclo más tarde.

La Generación de Dirección para la Memoria Intermedia 1 incluye por lo tanto un solo contador de dirección para cada una de las sub-memorias intermedias del rojo, el azul y el verde. La dirección inicial para RedAdr, BlueAdr y GreenAdr, es 0 al comienzo de cada línea en cada caso, y después de cada escritura en la Memoria Intermedia 1, la dirección se incrementa en 1, con reinicio cíclico en 36 ó 78, dependiendo de si se está escribiendo en la memoria intermedia el rojo, el verde o el azul. No todos los colores se escriben cada período de 75 ciclos. Una columna de verde requerirá típicamente el relleno al doble de velocidad que el rojo o el azul, por ejemplo.

La lógica se muestra en el pseudo-código que sigue:

20

Si el color a escribir es el Rojo
Escribir en Buffer1 del Rojo en RedAdr
RedAdr = RedAdr + 1 mod 36

O bien

Si el color a escribir es el Azul
Escribir en la Buffer1 del Azul en BlueAdr
BlueAdr = BlueAdr + 1 mod 36

O bien

Si el color a escribir es el Verde
Escribir en Buffer1 del Verde en GreenAdr
GreenAdr = GreenAdr + 1 mod 78

Endlf

25

30

35

40

45

10.2.11.2 *Generación de Dirección para RAM de Imagen*

Cada plano puede ser leído en un de dos orientaciones: girado en 0 o en 90 grados (giro anti-horario). Esto se traduce efectivamente como un acceso a modo de fila o a modo de columna a la imagen planar. Adicionalmente, permitimos la replicación de píxel marginal o el color constante para lecturas fuera de los límites de la imagen, así como también el reinicio cíclico de la imagen para formatos de impresión tales como Passport 31.

50

Al comienzo de cada línea de impresión debemos leer la ImageRAM 11 para cargar la Buffer1 114 tan rápido como sea posible. Esto equivale a un acceso simple a una muestra de cada ciclo. El remuestreo solamente puede ocurrir una vez que se han cargado 5 columnas, lo que significa 5 columnas de 6, 6, y 13 muestras, durante un total de 125 ciclos. Además de un ciclo extra para el valor final que ha de ser escrito en Buffer1 114 tras haberse cargado desde ImageRAM 11. Para hacer el conteo más fácil, redondeamos hasta 128 ciclos.

55

Tras los primeros 128 ciclos, se produce cada 75 ciclos la comprobación del requisito para cargar la siguiente columna de muestras para cada uno de los 3 colores, con las muestras apropiadas cargadas durante los 75 ciclos posteriores. Sin embargo, el establecimiento inicial de si se debe cargar durante el primer conjunto de 75 ciclos es siempre 1 para cada color. Esto permite que la 6ª columna final de cada color dentro de la Memoria Intermedia 1 sea rellena.

60

65

Al final de cada período de 75 ciclos, la banderola KernelXCarryOut de cada plano de color del Generador de Dirección de Función del Remuestreador 112 se comprueba para determinar si la siguiente columna de muestras debe

ES 2 282 136 T3

ser leída. De manera similar, un pulso AdvanceLine reinicia el proceso sobre la siguiente línea si KernelYCarryOut ha sido establecida.

5 Puesto que cada "lectura" comprende efectivamente 6 ó 13 lecturas para llenar una columna en la Memoria Intermedia 1, mantenemos una posición inicial con el fin de avanzar hasta la siguiente "lectura". También mantenemos un valor de coordenada para permitir la generación de coordenadas fuera-de-límites que permitan la replicación de píxel marginal, el color constante y el reinicio cíclico de imagen.

10 Consideramos la imagen 180 activa como que está dentro de unos límites particulares, con determinadas acciones que han de ser tomadas cuando las coordenadas están fuera del área activa. Las coordenada pueden ser o bien anteriores a la imagen, o bien estar dentro de la imagen, o bien posteriores a la imagen, tanto en términos de líneas como de píxeles. Esto se muestra en la Figura 67, aunque el espacio de fuera de la zona activa ha sido exagerado por motivos de claridad.

15 Obsérvese que debido a que utilizamos (0, 0) como inicio de la generación de coordenadas, MaxPixel y MaxLine son también conteos de píxel y de línea. Sin embargo, puesto que la generación de dirección se ejecuta desde las realizaciones de la función y los pulsos AdvanceLine desde la MJI 15, estos límites externos no son necesarios. La generación de dirección para una línea simplemente continúa hasta que se recibe el pulso AdvanceLine, y puede incluir replicación marginal, colores constantes para fuera de límites, o reinicio cíclico de píxel de imagen.

20 Si disponemos de una dirección, ADR, de la muestra actual, y deseamos movernos hasta la siguiente muestra, ya sea sobre la siguiente línea o ya sea sobre la misma línea, las coordenadas de la muestra cambiarán según se espera, pero la forma en que la dirección cambia depende de si estamos reiniciando cíclicamente la imagen activa, y debe producir replicación de píxel marginal cuando se precise.

25 Cuando no exista reinicio cíclico de la imagen (es decir, todos los formatos de impresión excepto Passport 31), realizamos las acciones de la Tabla 38 según avanzamos de línea o de píxel. Para girar una imagen en 90 grados, la CPU 10 simplemente cambia los valores Δ Line y Δ píxel.

30 Observando la Tabla 38, la única vez que ADR cambia es mediante Δ píxel cuando PixelSense es 0, y mediante Δ Line cuando LineSense es 0. Siguiendo estas simples normas, ADR será válida para replicación de píxel marginal. Por supuesto, si se desea un color constante para fuera de las coordenadas de los límites, se puede elegir ese valor en vez del valor almacenado en la dirección apropiada.

35 Para permitir el reinicio cíclico, comparamos simplemente el sentido anterior (-, 0, +) para Línea y Píxel con el nuevo sentido. Cuando el sentido es "-", utilizamos el avance según se describe en la Tabla 38, pero cuando la ordenada cae fuera de límites (es decir, moviéndose desde 0 hasta +), actualizamos la ADR

TABLA 38

Acciones a Llevar a Cabo cuando se Avanza en Píxel o Línea

Línea ^a	Píxel ^b	Cambio de píxel	Cambio de Línea
-	-		
-	0	$Adr = Adr + \Delta píxel$	
-	+		
0	-		$Adr = Adr + \Delta Line$
0	0	$Adr = Adr + \Delta píxel$	$Adr = Adr + \Delta Line$
0	+		$Adr = Adr + \Delta Line$
+	-		
+	0	$Adr = Adr + \Delta píxel$	
+	+		

45 a... Comparamos la ordenada de Línea actual con ActiveStartLine y ActiveEndLine
 Si $Línea < ActiveStartLine$, llamamos al valor "-".
 Si $ActiveStartLine \leq Línea \leq ActiveEndLine$, llamados al valor "0".
 Si $ActiveEndLine \leq Línea$, llamamos al valor "+".

60 b... Comparamos la ordenada de Píxel actual con ActiveStartPixel y ActiveEndPixel.
 Si $Píxel < ActiveStartPixel$, llamamos al valor "-".
 Si $ActiveStartPixel \leq Píxel \leq ActiveEndPixel$, llamamos al valor "0"
 Si $ActiveendPixel \leq Píxel$, llamamos al valor "+"

ES 2 282 136 T3

con un nuevo valor que no está basado en un delta. Suponiendo que mantenemos la dirección inicial para la línea actual, de modo que podemos avanzar hasta el inicio de la siguiente línea una vez que la línea actual ha sido generada, podemos hacer lo siguiente:

- 5 • Si el cambio es en Píxel, y el sentido del píxel cambia desde 0 a + (lo que indica que nos hemos ido más allá del borde de la imagen), sustituimos Adr por LineStartAdr y sustituimos Píxel por ActiveStartPixel. Línea se mantiene igual.
- 10 • Si el cambio es en Línea, y el sentido cambia desde 0 a + (lo que indica que nos hemos ido más allá del borde de la imagen), restamos DeltaColumn de Adr, y sustituimos Línea por ActiveStartLine. Píxel se mantiene igual. Deltacolumn es la desviación de dirección para generar las direcciones de (Píxel, ActiveStartLine) desde (Píxel, ActiveEndLine-1).

15 La lógica para cargar el número de muestras del conjunto (ya sea 6 ó 13, dependiendo del color), se muestra en el pseudo-código que sigue.

```
20       línea = FirstSampleLine
       píxel = FirstSamplePixel
       adr = FirstSampleAdr
25       Hacerlo N veces (6 ó 13)
          oldPixelSense = PixelSense(píxel)
          oldLineSense = LineSense(gLine)
30       inActive = (oldLineSense == InActive) Y (oldPixelsense == InActive)
          Si (NO (inactive) Y UseConstant)
              Muestra = ConstantColor
35       o bien
          Muestra = Fetch(adr)
40       EndIf
          línea = línea + 1
          Si ((LineSense(line) == "+") Y wrapImage)
45            adr = adr - DeltaColumn
              línea = ActiveStartLine
          Elseif ((LineSense(line) == "0") Y ((oldLineSense == "0")))
50            adr = adr + DeltaLine
          EndIf
55       End Do
```

60 El establecimiento de tales variables como FirstSampleLine, FirstSamplePixel y FirstSampleAdr se realiza en la sección de generador de dirección que corresponda para realizar banderolas desde el Generador de Dirección de Función, así como también pulsos AdvanceLine desde la MJI. La lógica para esta parte de la generación de dirección se muestra en el siguiente pseudo-código:

65

```

FirstSamplePixel = 0
5 FirstSampleLine = 0

```

```

10 FirtsAmpleAdr = FirstSampleAdr = ActiveStartAddress
Conteo = 0
Hacer siempre
15 Si ((KernelXCarryOut) O (AdvanceLine Y KernelYCarryOut) o (conteo < 5))
    Hacer N Muestras para este plano de color (véase el pseudo-código anterior)
    Endif
20 oldPixelSense = PixelSense(FirstSamplePixel)
    oldLineSense = LineSense(FirstSampleLine)
    Si (AdvanceLine Y KernelYCarryOut)
25         count = 0
            FirstSampleLine = FirstSampleLine + 1
            FirstSamplePixel = 0
30         Si ((LineSense(FirstSampleLine) == "+") Y wrapImage)
                FirstLineSampleAdr = StartAddress
                FirstSampleLine = ActiveStartLine
35         Elseif ((LineSense(FirstSampleLine) == "0") Y (oldLineSense == "0"))
                FirstLineSampleAdr = FirstLineSampleAdr + DeltaLine
40         Endif
            FirstSampleAdr = FirstLineSampleAdr
    Elseif (KernelXCarryOut O (count < 5))
45         FirstSamplePixel = FirstSamplePixel + 1
            count = count + 1
            Si ((PixelSense(FirstSamplePixel) == "+") Y wrapImage)
50                 FirstSampleAdr = FirstLineSampleAdr
                    FirstSamplePixel = ActiveStartPixel
            Elseif ((PixelSense(FirstSamplePixel)) == "0") Y (oldPixelSense == "0"))
55                 FirstSampleAdr = FirstSampleAdr + DeltaPixel
            Endif
60         Endif
    Endif

```

65

ES 2 282 136 T3

10.2.11.3 Sumario de Registro

Existe un número de registros que deben ser determinados con anterioridad a la impresión de una imagen. Éstos se resumen aquí en la Tabla 39. Para girar una imagen en 90 grados, simplemente se intercambian los valores DeltaLine y DeltaPixel, y se proporciona un nuevo valor DeltaColumn.

TABLA 39

Registros que se Requiere que sean Determinados mediante el Llamador antes de la Impresión

Nombre del Registro	Descripción
Parámetros de Acceso de Imagen	
WrapImage	El mosaico lee la imagen para replicar la imagen cuando está fuera de los límites de imagen
UseConstant	Si es 0, la replicación marginal o el reinicio cíclico de la imagen ocurre con la lectura de los límites de imagen Si es 1, se devuelve un color constante
Rojo	
ActiveStartAddressR	La dirección de la muestra roja (ActiveStartPixel, ActiveStartLine) en ImageRAM
ActiveStartLineR	La primera línea válida para la imagen en el espacio del rojo (con relación a la línea 0)
ActiveEndLineR	La primera línea fuera de los límites de la imagen en el espacio del rojo
ActiveStartPixelR	El primer píxel válido en el espacio del rojo (con relación al píxel 0)
ActiveEndPixelR	El primer píxel fuera de los límites de la imagen en el espacio del rojo
DeltaLineR	La cantidad a añadir a la dirección actual para moverse desde una línea hasta la siguiente en el espacio del rojo
DeltaPixelR	La cantidad a añadir a la dirección actual para moverse desde un píxel hasta el siguiente de la misma línea en el espacio del rojo
DeltaColumnR	La cantidad a añadir a la dirección actual para moverse desde un píxel de la última línea de la zona de imagen Active hasta el mismo píxel de la primera línea de la zona de imagen Active en el espacio del rojo
ConstantcolorR	Valor de color rojo a utilizar si la dirección está fuera de límites y UseConstant = 1
Verde	
ActiveStartAddressG	La dirección de la muestra verde (ActiveStartPixel, ActiveStartLine) en ImageRAM
ActiveStartLineG	La primera línea válida para la imagen en el espacio del verde (con relación a la línea 0)
ActiveEndLineG	La primera línea fuera de límites para la imagen en el espacio del verde
ActiveStartPixelG	El primer píxel válido para la imagen en el espacio del verde (con relación al píxel 0)
ActiveEndPixelG	El primer píxel fuera de límites para la imagen en el espacio del verde
DeltaLineG	La cantidad a añadir a la dirección actual para moverse desde una línea hasta la siguiente en el espacio del verde

5	DeltaPixelG	La cantidad a añadir a la dirección actual para moverse desde un píxel hasta el siguiente de la misma línea en el espacio del verde
	DeltaColumnG	La cantidad a añadir a la dirección actual para moverse desde un píxel de la última línea de la zona de imagen Active hasta el mismo píxel de la primera línea de la zona de imagen Active en el espacio del verde
10	ConstantColorG	Valor del color verde que se ha de usar si la dirección está fuera de límites y UseConstant = 1
	Azul	
15	ActiveStartAdresB	La dirección de la muestra azul (ActiveStartPixel, ActiveStartLine) en ImageRAM
	ActiveStartLineB	La primera línea válida para la imagen en el espacio del azul (con relación a la línea 0)
20	ActiveEndLineB	La primera línea fuera de límites para la imagen en el espacio del azul
	ActiveStartPixelB	El primer píxel válido para la imagen en el espacio del azul (con relación al píxel 0)
25	ActiveEndPixelB	El primer píxel fuera de límites para la imagen en el espacio del azul
	DeltaLineB	La cantidad a añadir a la dirección actual para moverse desde una línea hasta la siguiente en el espacio del azul
30	DeltaPixelB	La cantidad a añadir a la dirección actual para moverse desde un píxel hasta el siguiente sobre la misma línea en el espacio del azul
	DeltaColumnB	La cantidad a añadir a la dirección actual para moverse desde un píxel de la última línea de la zona de imagen Active hasta el mismo píxel de la primera línea de la zona de imagen Active en el espacio del azul
35	ConstantColorB	Valor de color azul a utilizar si la dirección está fuera de límites y UseConstant = 1
40	Parámetros de Balance del Blanco y de Expansión de Rango	
	RedLowThreshold	Valor de 8 bits sustraído de los valores de entrada del rojo
	GreenLowThreshold	Valor de 8 bits sustraído de los valores de entrada del verde
45	BlueLowThreshold	Valor de 8 bits sustraído de los valores de entrada del azul
	RedScaleFactor	Factor escalar de 8 bits utilizado para la expansión de rango de los píxeles rojos
	GreenScaleFactor	Factor escalar de 8 bits utilizado para la expansión de rango de los píxeles verdes
50	BlueScaleFactor	Factor escalar de 8 bits utilizado para la expansión de rango de los píxeles azules

11 Referencias

[1] Silverbrook Research, 1998, Authentication of Consumables

[2] Silverbrook Research, 1998, Authentication Chip.

Aunque la invención ha sido descrita con referencia a los ejemplos específicos, los expertos en la materia podrán apreciar que se puede materializar de otras muchas formas. Los párrafos numerados que siguen proporcionan las direcciones con una mejor indicación del alcance de la invención, aunque otras características novedosas y combinación de características aparecerán también claramente a partir de la descripción que sigue.

REIVINDICACIONES

5 1. Un procedimiento para proporcionar una imagen para impresión a una resolución de punto bi-nivel predeterminada que corresponde a una resolución de tono continuo predeterminada, incluyendo el procedimiento las etapas de:

10 recibir un primer conjunto de datos indicativos de la imagen, estando el primer conjunto de datos en formato Bayer planarizado que comprende diferentes planos (45, 46, 47) de color, teniendo al menos uno (46) de los planos de color una resolución diferente a la del otro plano o los otros planos (45, 47) de color;

15 realizar reconstrucción y remuestreo (64) de imagen de cada plano de color del primer conjunto de datos, para crear un segundo conjunto de datos de resolución de tono continuo predeterminada que sea mayor que la resolución de cada uno de los planos de color del primer conjunto de datos;

20 convertir (67) el segundo conjunto de datos en un tercer conjunto de datos de la resolución de punto bi-nivel predeterminada que sea mayor que la resolución de tono continuo predeterminada del segundo conjunto de datos, y

25 hacer que el tercer conjunto de datos se encuentre disponible para una impresora (69) a la resolución de punto bi-nivel predeterminada.

30 2. Un procedimiento de acuerdo con la reivindicación 1, en el que el primer conjunto de datos está en formato de rojo, verde y azul RGB, y la impresora actúa en respuesta a un formato de cyan, magenta y amarillo CMY, incluyendo el procedimiento la etapa de convertir (66) el segundo conjunto de datos desde un formato RGB a un formato CMY.

35 3. Un procedimiento de acuerdo con la reivindicación 1, que incluye la etapa de intensificar (65) el segundo conjunto de datos.

40 4. Un procedimiento de acuerdo con la reivindicación 1, en el que el primer conjunto de datos se obtiene a partir de un dispositivo sensor, y el procedimiento incluye la etapa de compensar el primer conjunto de datos respecto a las no linealidades en el dispositivo sensor.

45 5. Un procedimiento de acuerdo con la reivindicación 4, en el que la etapa de compensación incluye convertir el primer conjunto de datos desde una pluralidad de "x" muestras de bits a una pluralidad de "y" muestras de bits, siendo $x > y$.

50 6. Un procedimiento de acuerdo con la reivindicación 5, en el que $x = 10$ e $y = 8$.

55 7. Un procedimiento de acuerdo con la reivindicación 1, que incluye las etapas adicionales de:

determinar respecto al primer conjunto de datos, el m% de píxeles más oscuros y el n% de píxeles con más luminosidad;

60 ajustar el primer conjunto de datos para igualar el m% de píxeles más oscuros, y

ajustar el primer conjunto de datos para igualar el n% de píxeles con más luminosidad.

65 8. Un procedimiento de acuerdo con la reivindicación 1, que incluye la etapa adicional de ajustar el primer conjunto de datos para proporcionar un balance (62) del blanco predeterminado.

9. Un procedimiento de acuerdo con la reivindicación 1, que incluye la etapa adicional de ajustar el primer conjunto de datos para proporcionar una expansión (63) de rango predeterminada.

70 10. Un procedimiento de acuerdo con la reivindicación 8 o la reivindicación 9, en el que la resolución de cada plano de color del primer conjunto de datos se incrementa mientras que se mantiene la misma resolución espacial.

75 11. Un procedimiento de acuerdo con la reivindicación 1, en el que el primer conjunto de datos se ajusta selectivamente para que proporcione la imagen con una orientación rotacional predeterminada.

80 12. Un aparato para proporcionar una imagen para impresión a una resolución de punto bi-nivel predeterminada que corresponde con una resolución de tono continuo predeterminada, incluyendo el aparato:

85 medios de entrada para recibir un primer conjunto de datos indicativos de la imagen, estando el primer conjunto de datos en formato Bayer planarizado que comprende diferentes planos (45, 46, 47) de color, siendo al menos uno (46) de los planos de color de una resolución diferente a la del otro plano o los otros planos (46, 47) de color,

ES 2 282 136 T3

medios de muestreo para reconstrucción y remuestreo (64) de imagen del primer conjunto de datos, para crear un segundo conjunto de datos con una resolución de tono continuo predeterminada que sea mayor que la resolución de cada uno de los planos de color del primer conjunto de datos,

5 medios de procesamiento para convertir (67) el segundo conjunto de datos en un tercer conjunto de datos con la resolución de punto bi-nivel predeterminada que es mayor que la resolución de tono continuo predeterminada del segundo conjunto de datos, y

10 medios de salida para hacer que el tercer conjunto de datos se encuentre disponible (69) para que una impresora imprima a la resolución de punto bi-nivel predeterminada.

13. Un aparato de acuerdo con la reivindicación 12, en el que el primer conjunto de datos está en formato RGB de rojo, verde y azul, y la impresora actúa en respuesta a un formato CMY de cyan, magenta y amarillo, convirtiendo (66) los medios de procesamiento el segundo conjunto de datos desde un formato RGB a un formato CMY.

15 14. Un aparato de acuerdo con la reivindicación 12, que incluye además un filtro de intensificación para intensificar (65) el segundo conjunto de datos.

20 15. Un aparato de acuerdo con la reivindicación 12, en el que el primer conjunto de datos se obtiene a partir de un dispositivo sensor, y el medio de entrada compensa el primer conjunto de datos en cuanto a las no linealidades en el dispositivo sensor.

25 16. Un aparato de acuerdo con la reivindicación 15, en el que la compensación respecto a las no linealidades incluye convertir el primer conjunto de datos desde una pluralidad de "x" muestras de bits a una pluralidad de "y" muestras de bits, siendo $x > y$.

17. Un aparato de acuerdo con la reivindicación 16, en el que $x = 10$ e $y = 8$.

30 18. Un aparato de acuerdo con la reivindicación 12, en el que los medios de entrada:
determinan con relación al primer conjunto de datos, el m% de píxeles más oscuros y el n% de píxeles más brillantes;

35 ajustan el primer conjunto de datos para igualar el m% de píxeles más oscuros, y

ajustan el primer conjunto de datos para igualar el n% de píxeles más brillantes.

40 19. Un aparato de acuerdo con la reivindicación 12, en el que los medios de entrada ajustan el primer conjunto de datos para proporcionar un balance (62) del blanco predeterminado.

20. Un aparato de acuerdo con la reivindicación 12, en el que los medios de entrada ajustan el primer conjunto de datos para proporcionar una expansión (63) de rango predeterminada.

45 21. Un aparato de acuerdo con la reivindicación 19 o la reivindicación 20, en el que los medios de entrada incrementan la resolución de cada plano de color del primer conjunto de datos mientras que se mantiene la misma resolución espacial.

50 22. Un aparato de acuerdo con la reivindicación 12, en el que los medios de entrada ajustan selectivamente el primer conjunto de datos para proporcionar la imagen con una orientación rotacional predeterminada.

23. Una cámara que incluye:

un conjunto CCD para proporcionar una imagen Bayer;

55 una impresora para proporcionar selectivamente una imagen impresa, y

un aparato de acuerdo con la reivindicación 12, para recibir la imagen Bayer y proporcionar a la impresora el tercer conjunto de datos de tal modo que se produzca la imagen impresa.

60

65

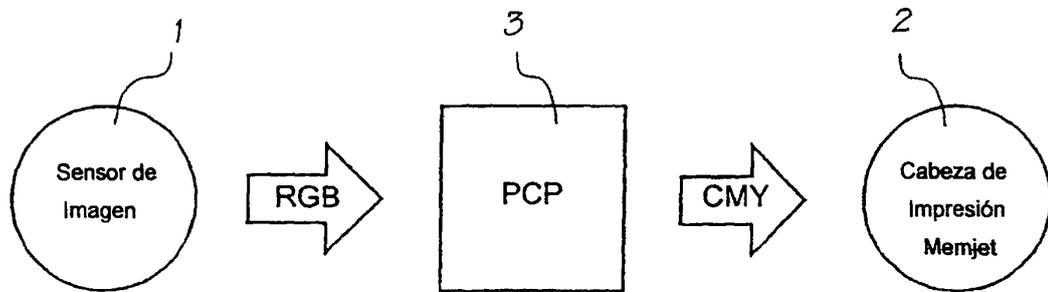


FIG. 1

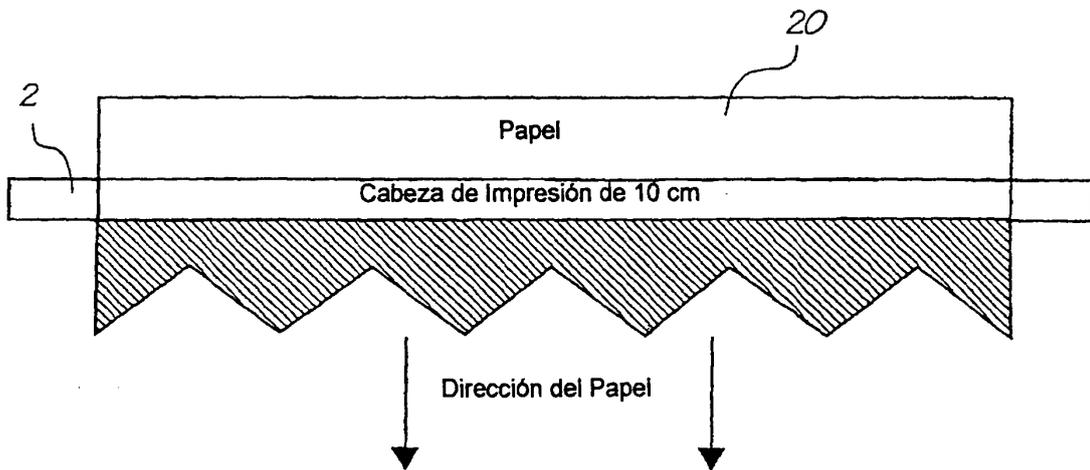


FIG. 4

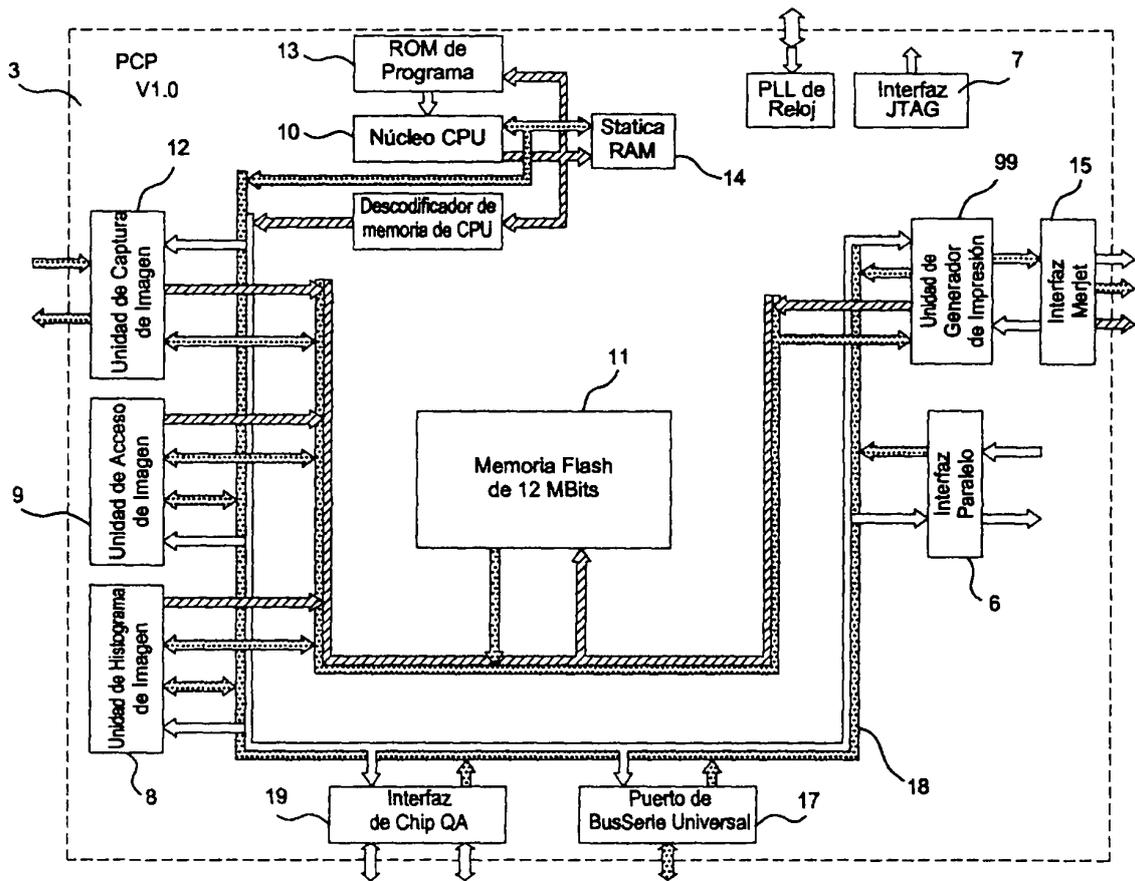


FIG. 2

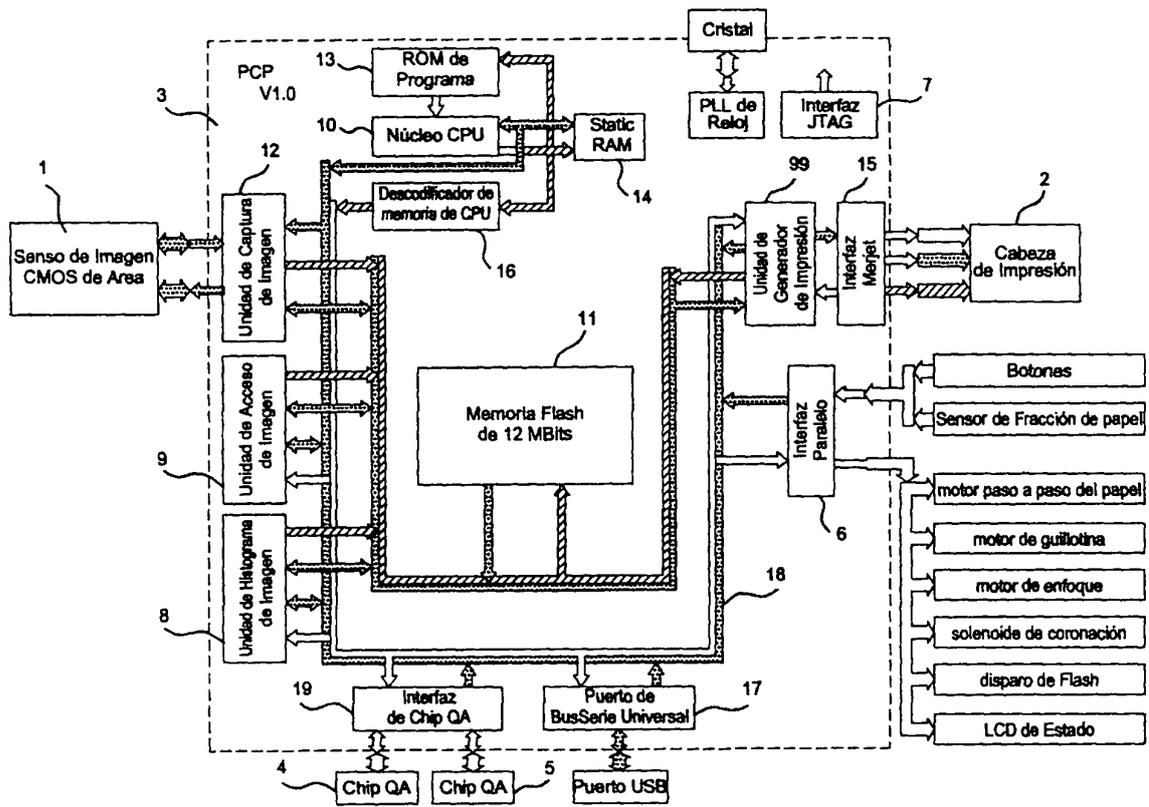
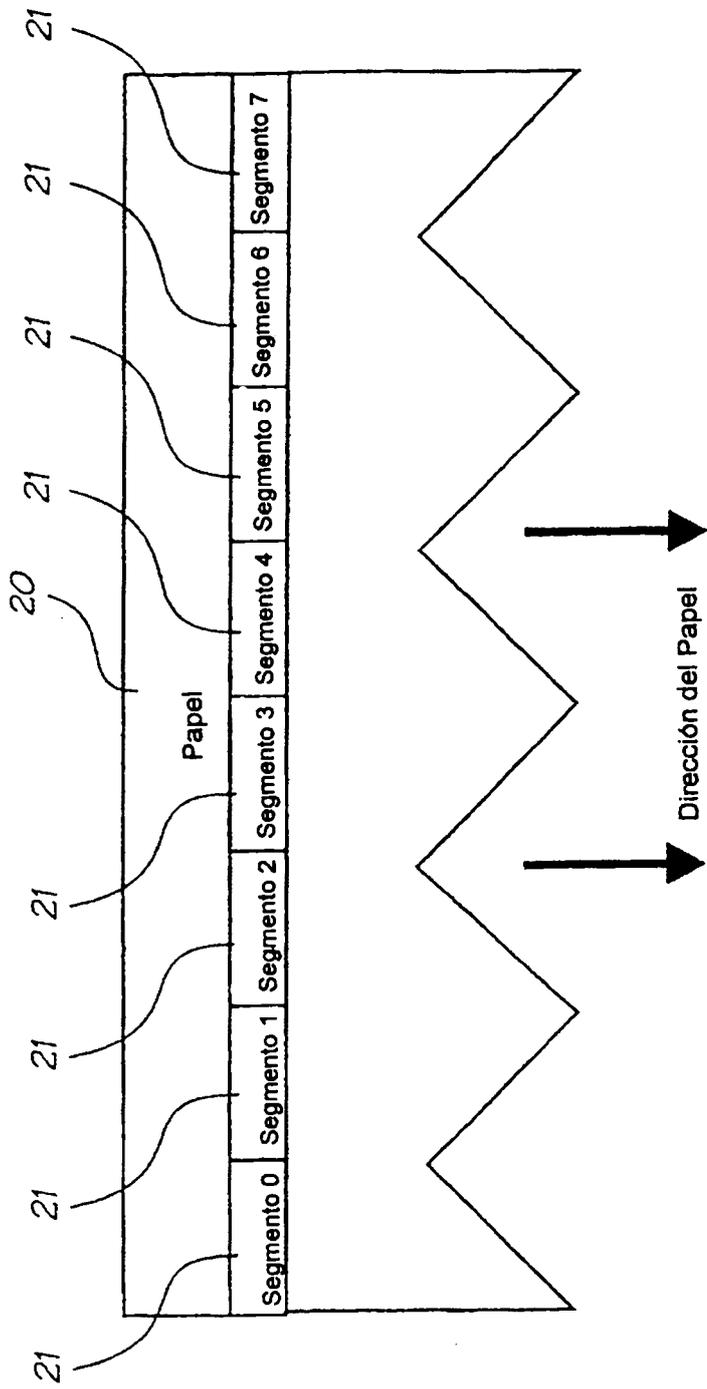


FIG. 3



1 Segmento
= 0.5 inch
= 12700 μm

FIG. 5

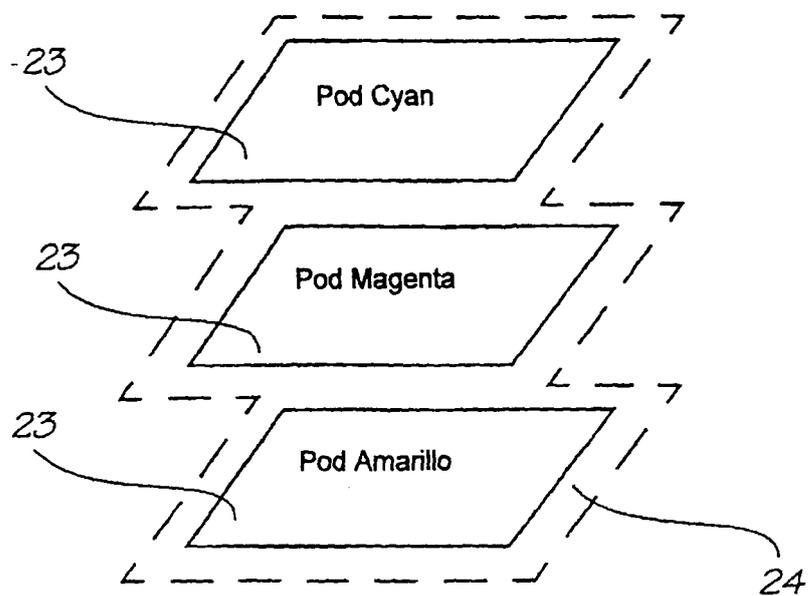
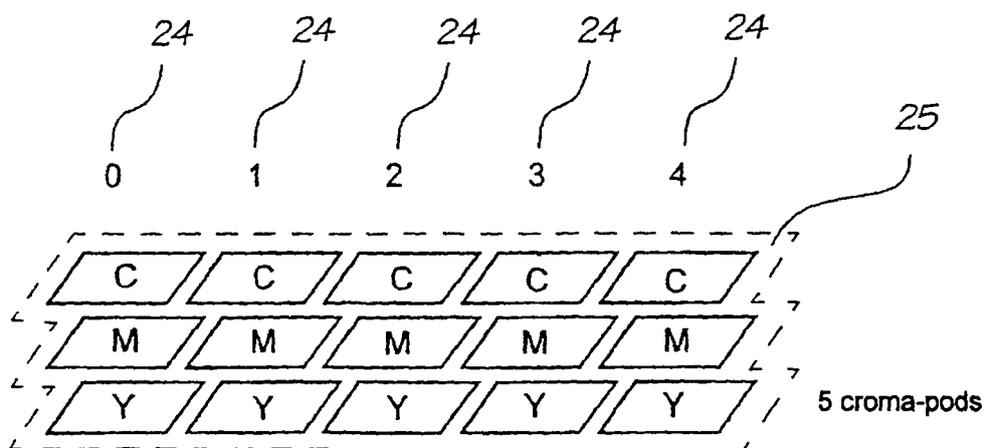


FIG. 8



C = Cyan
M = Magenta
Y = Amarillo

FIG. 9

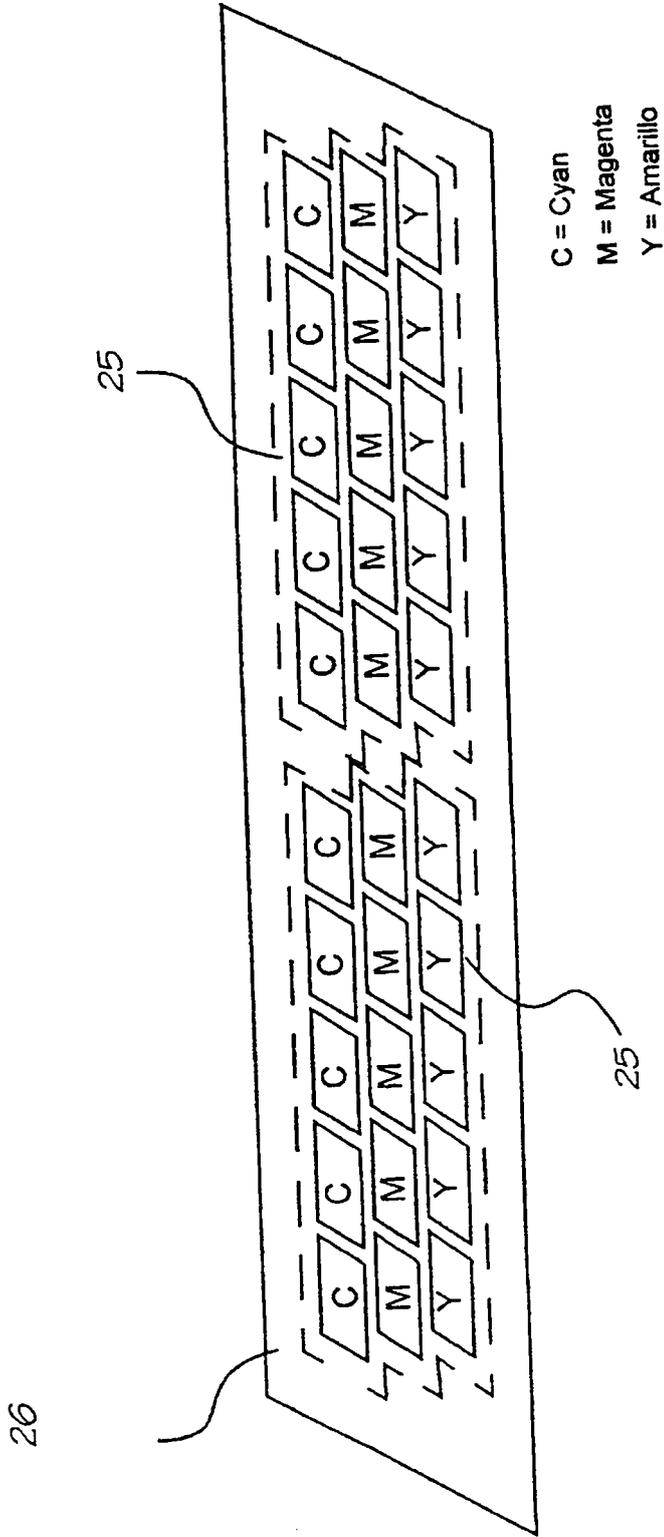


FIG. 10

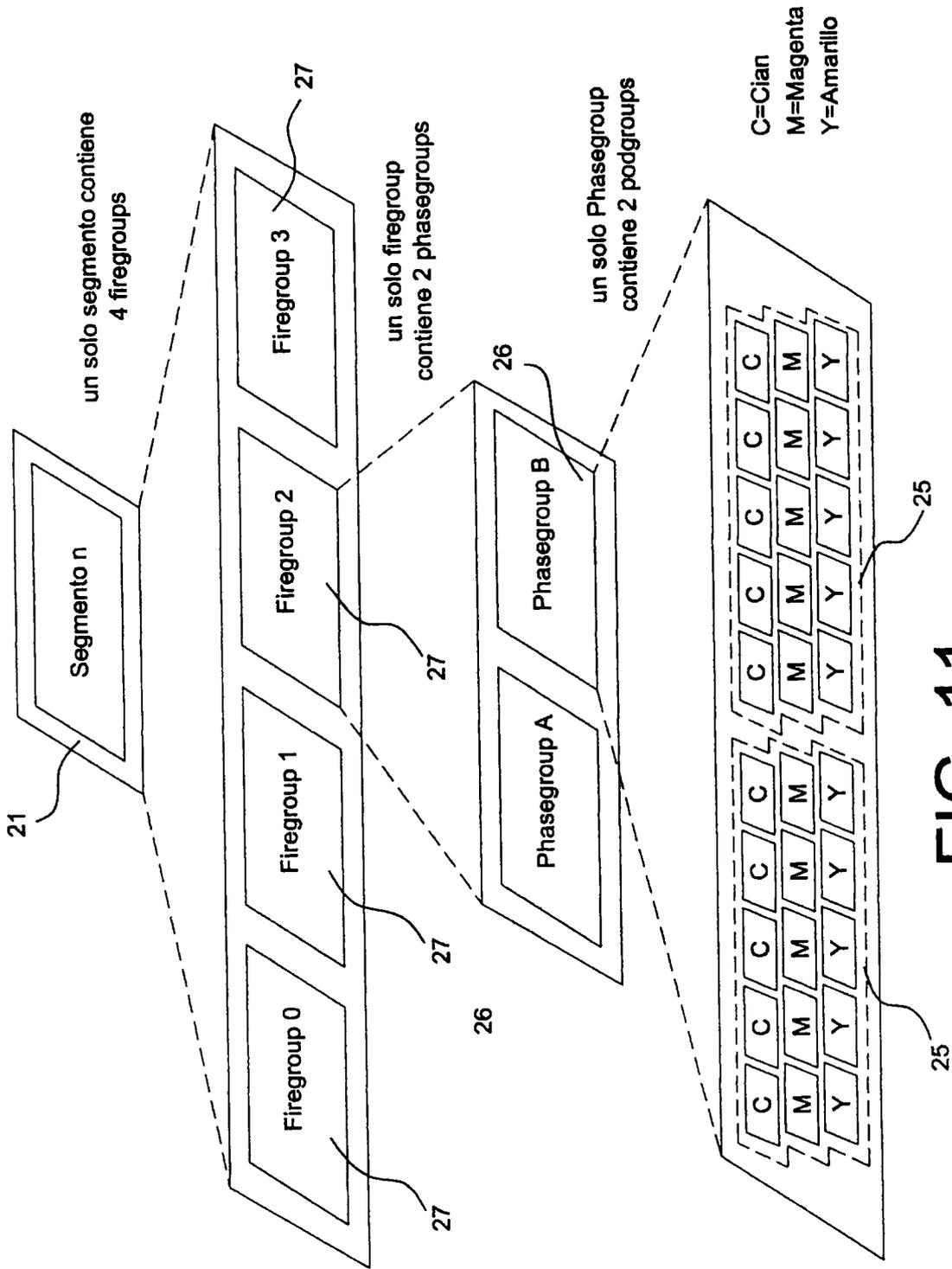


FIG. 11

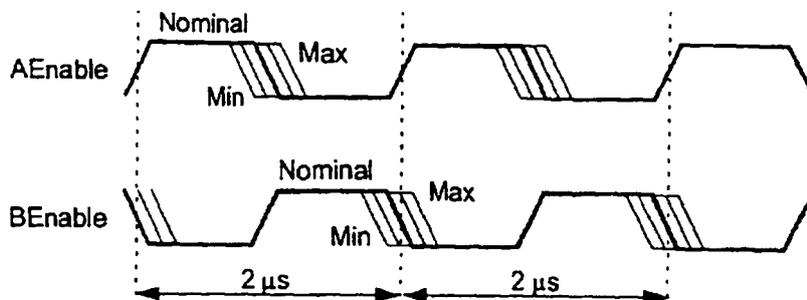


FIG. 12

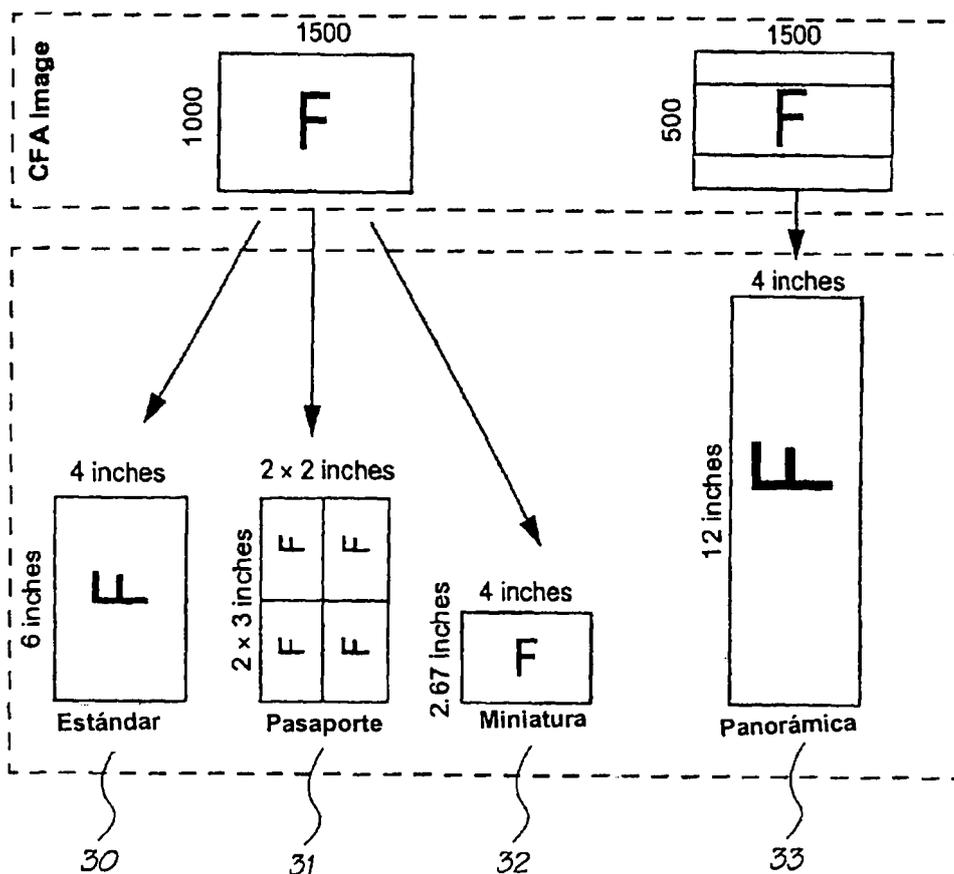


FIG. 13

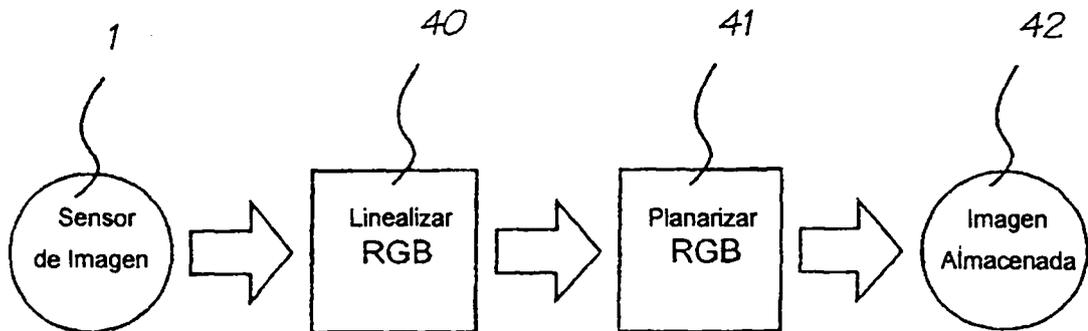


FIG. 14

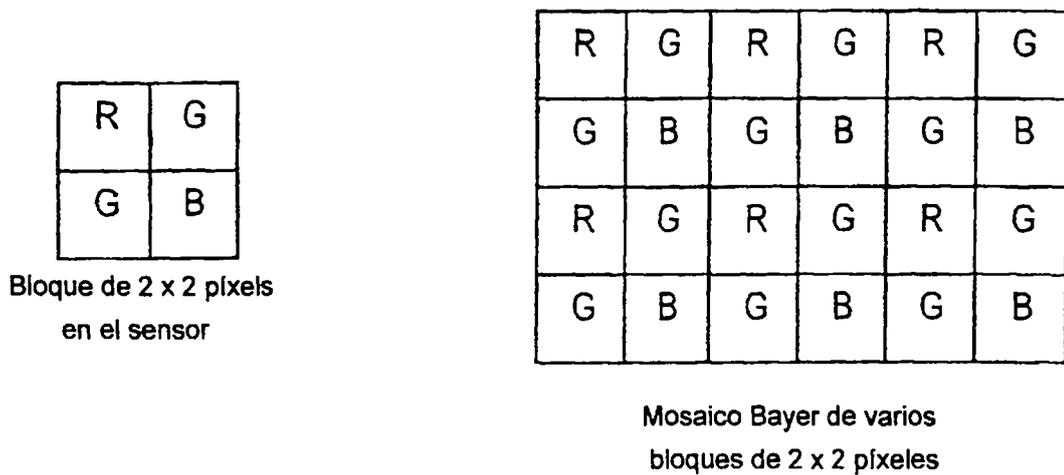


FIG. 15

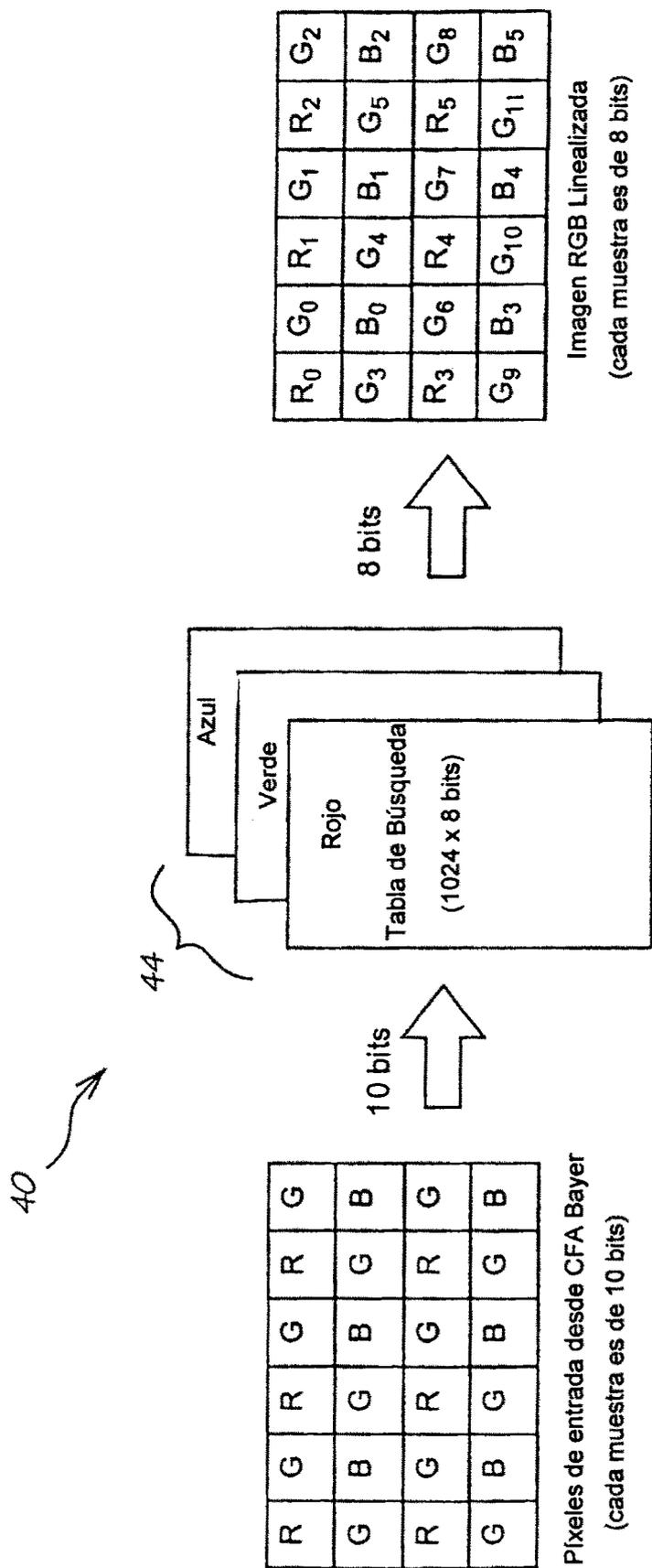
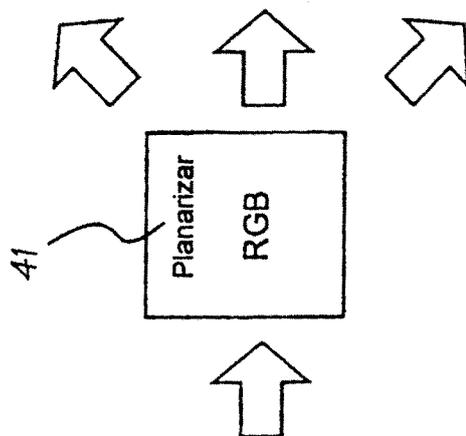
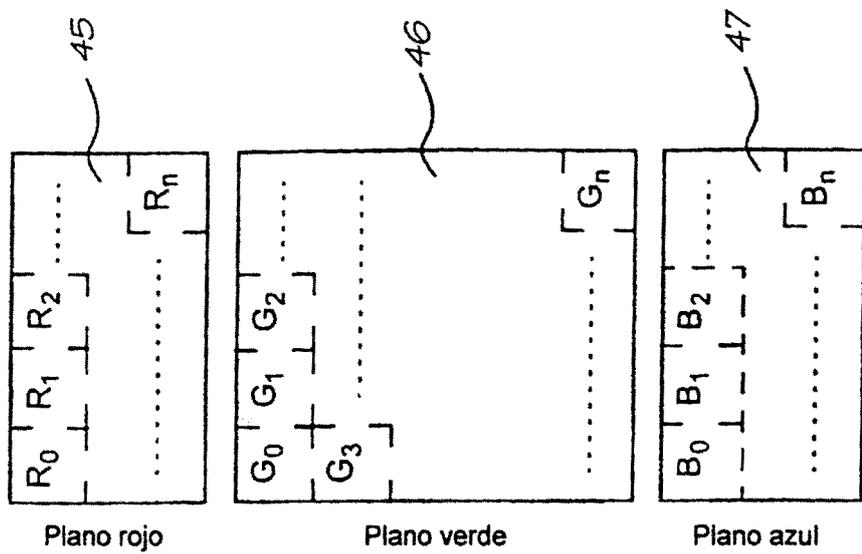


FIG. 16



R ₀	G ₀	R ₁	G ₁	R ₂	G ₂
G ₃	B ₀	G ₄	B ₁	G ₅	B ₂
R ₃	G ₆	R ₄	G ₇	R ₅	G ₈
G ₉	B ₃	G ₁₀	B ₄	G ₁₁	B ₅

Imagen RGB linealizada
(cada muestra es de 8 bits)

FIG. 17

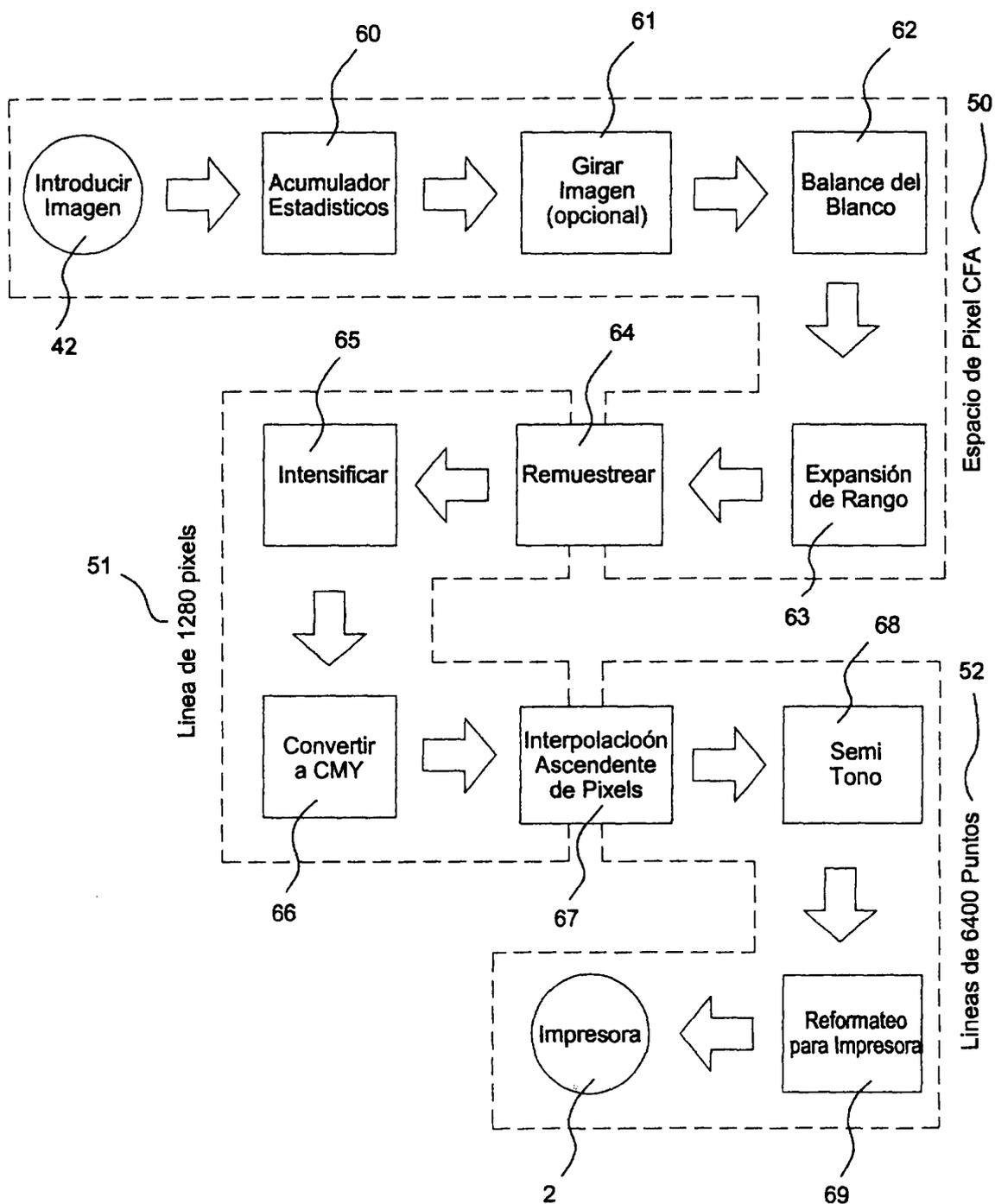


FIG. 18

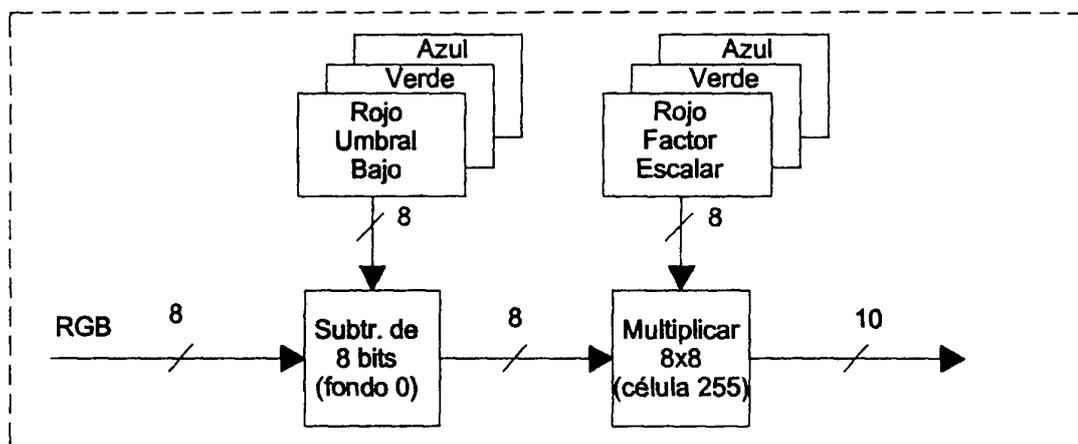
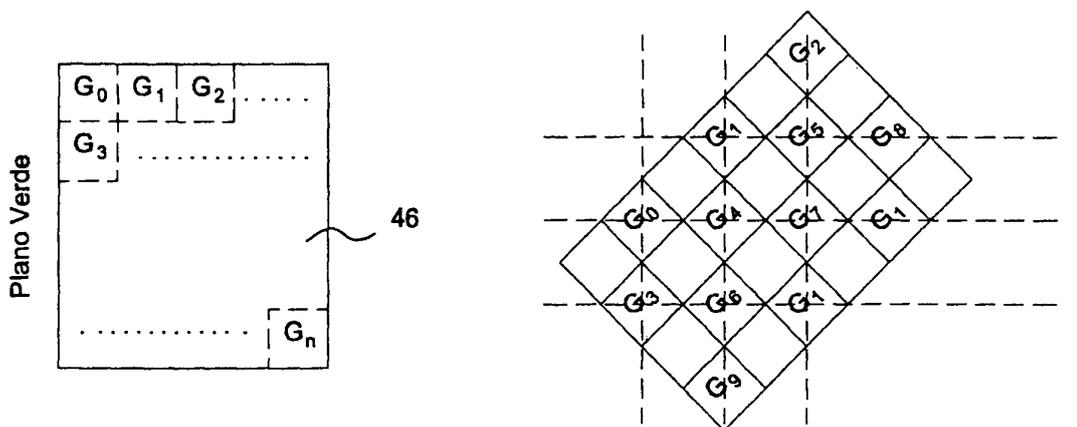


FIG. 21



Plano Verde res CFA Completo
(los pixels capturados son marcados)

FIG. 23

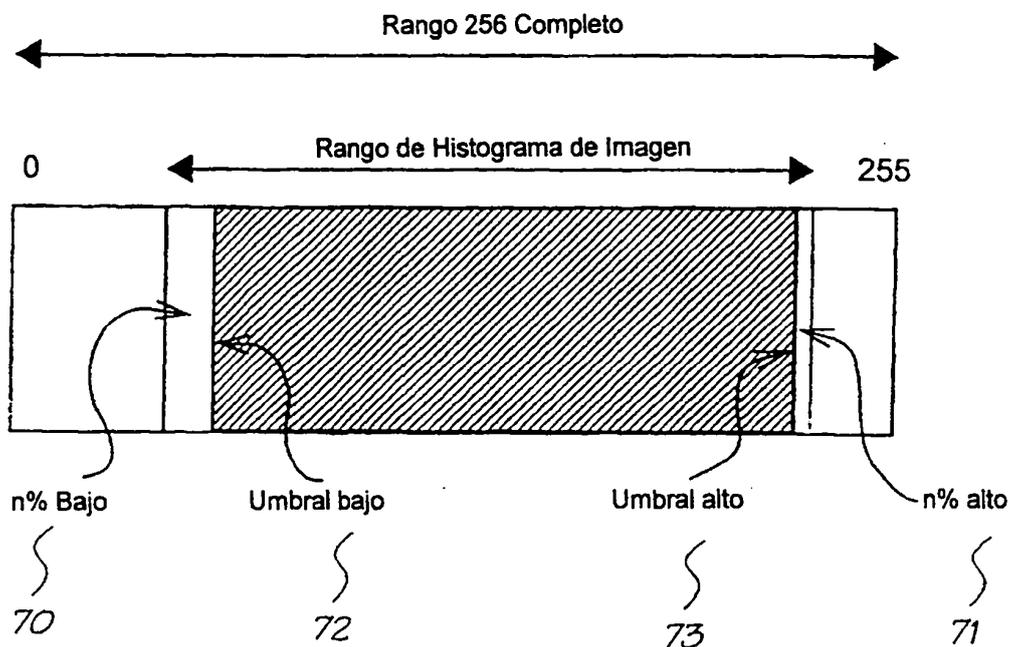


FIG. 19

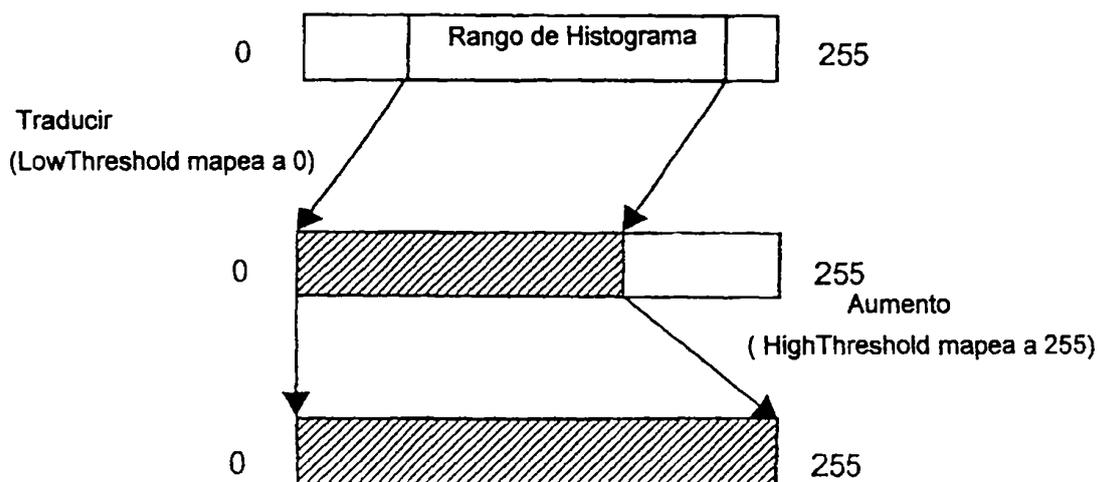


FIG. 20

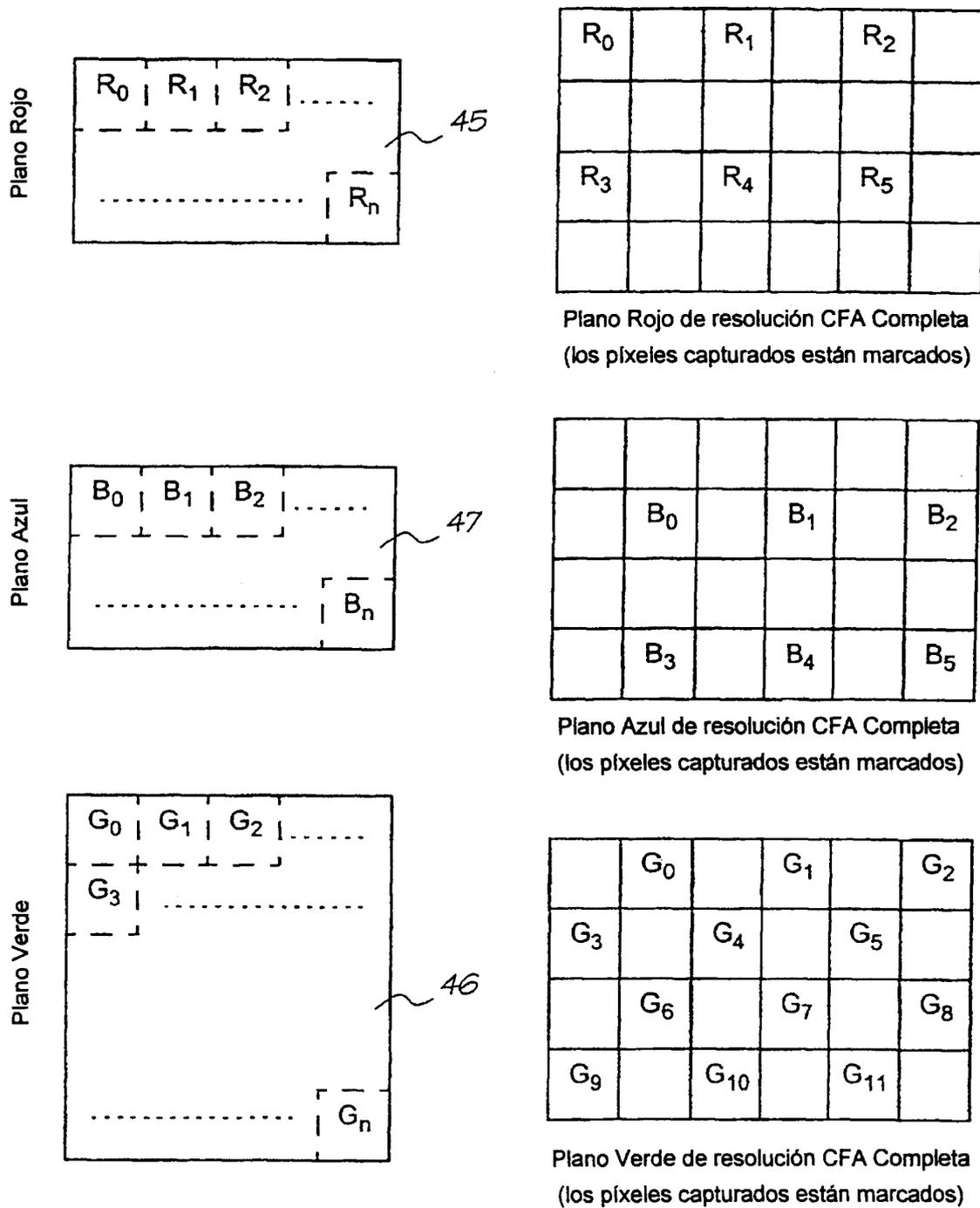


FIG. 22

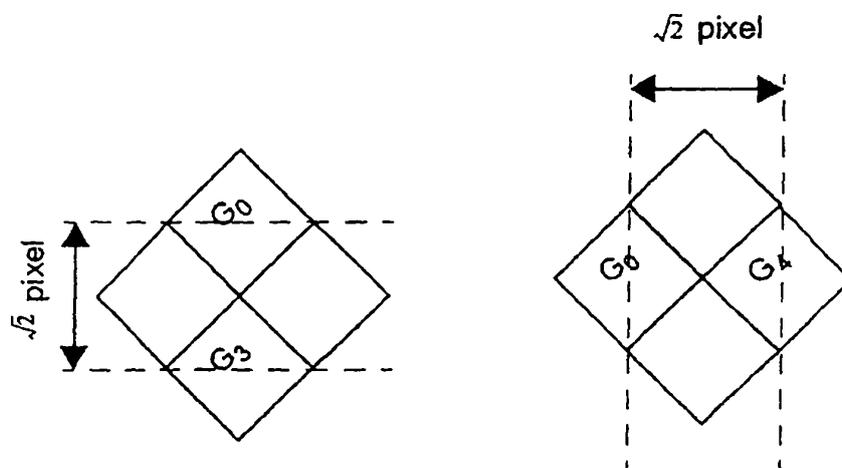


FIG. 24

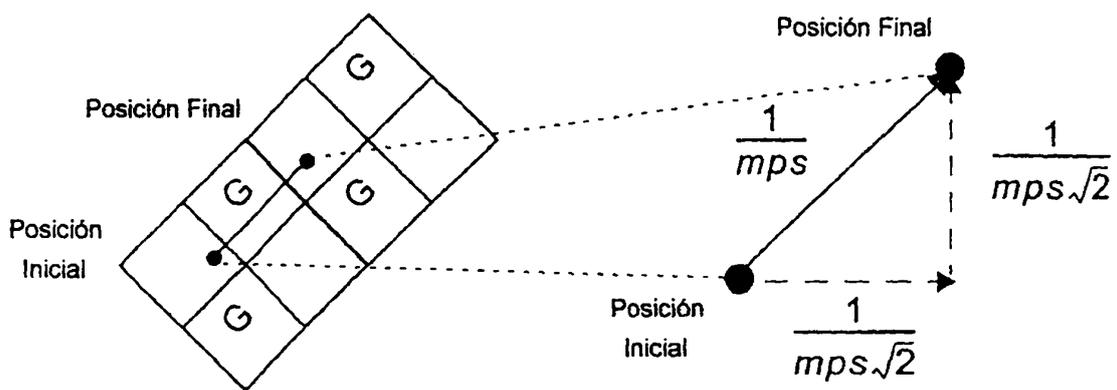


FIG. 25

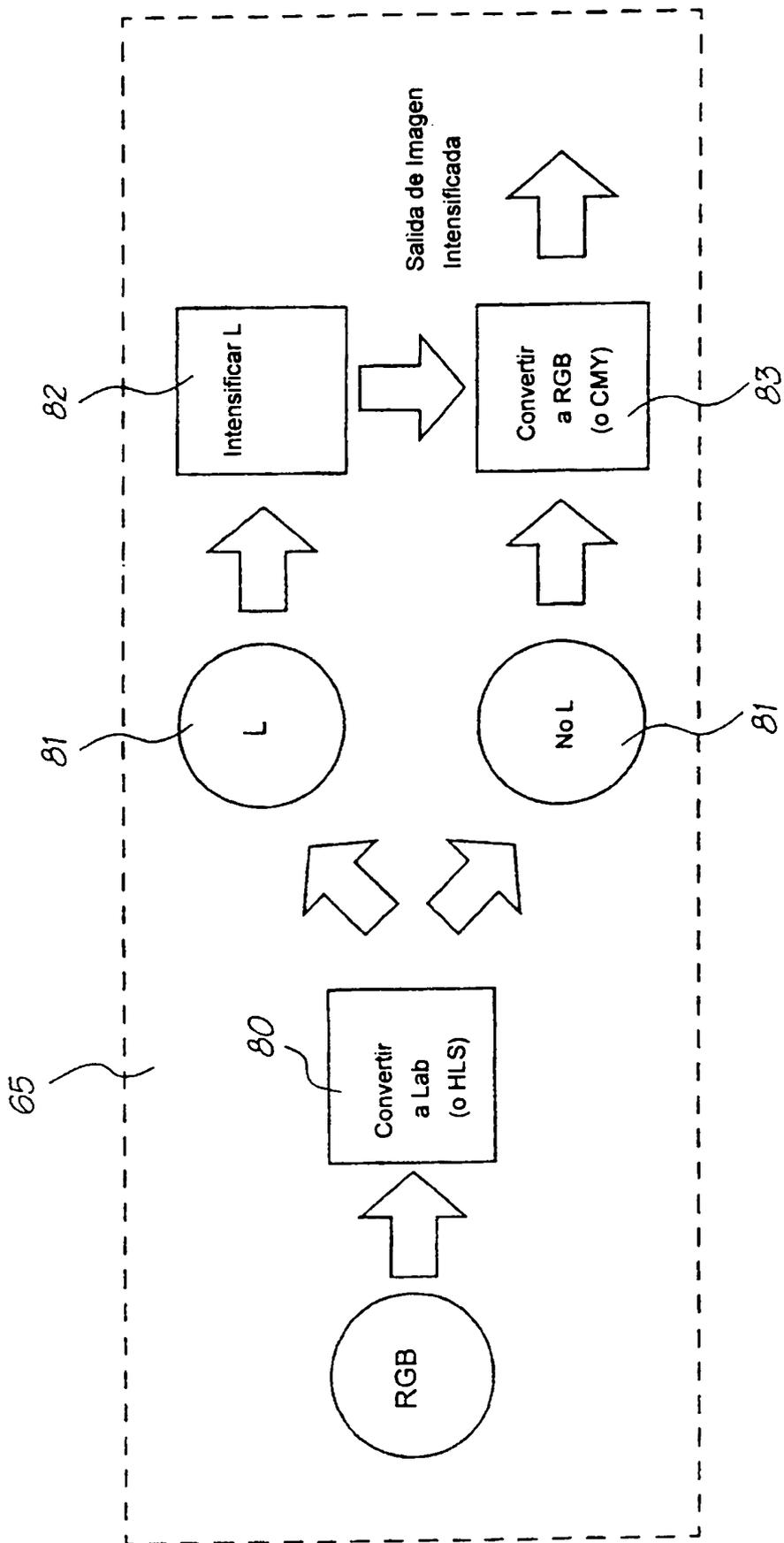


FIG. 26

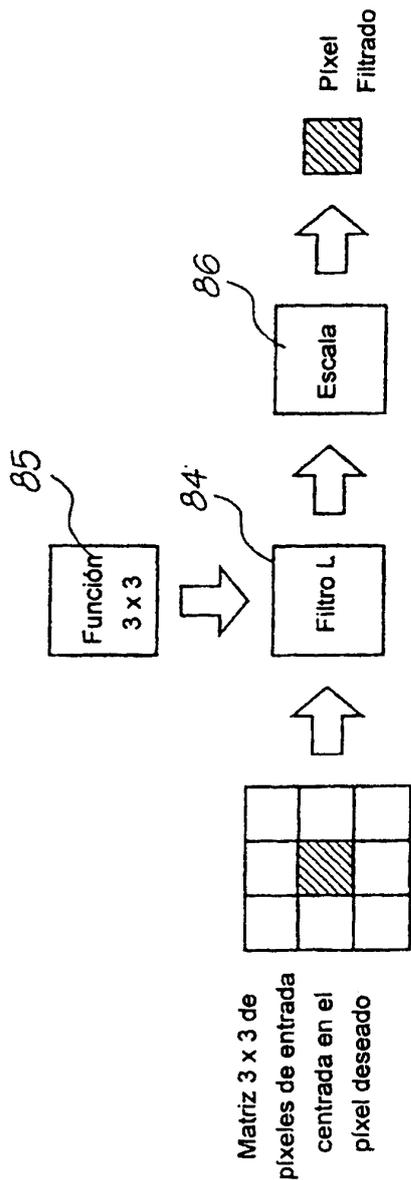


FIG. 27

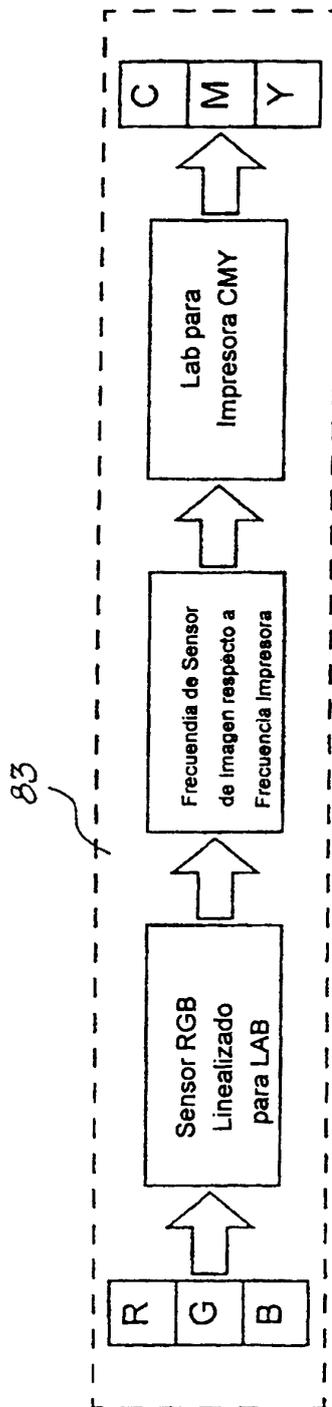


FIG. 28

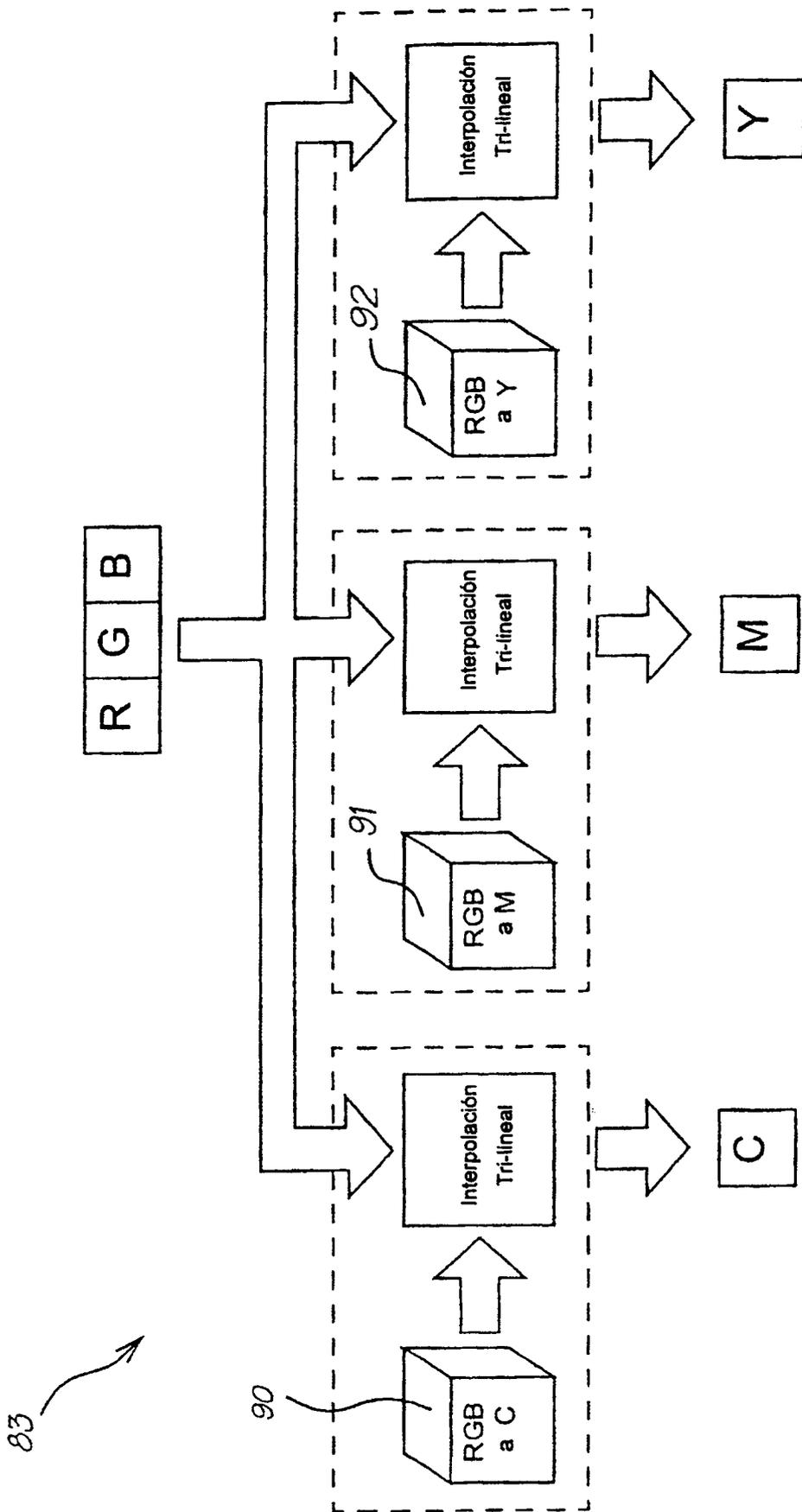


FIG. 29

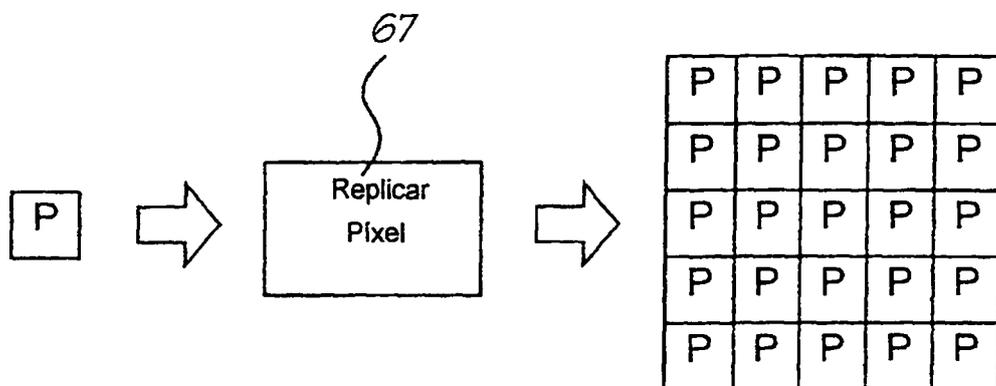


FIG. 30

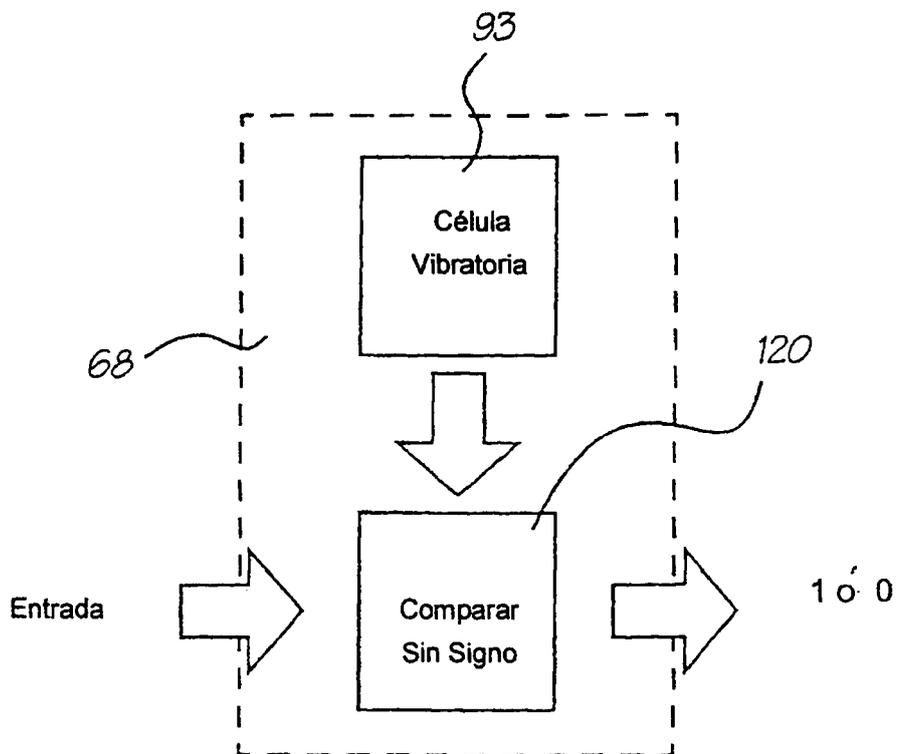


FIG. 31

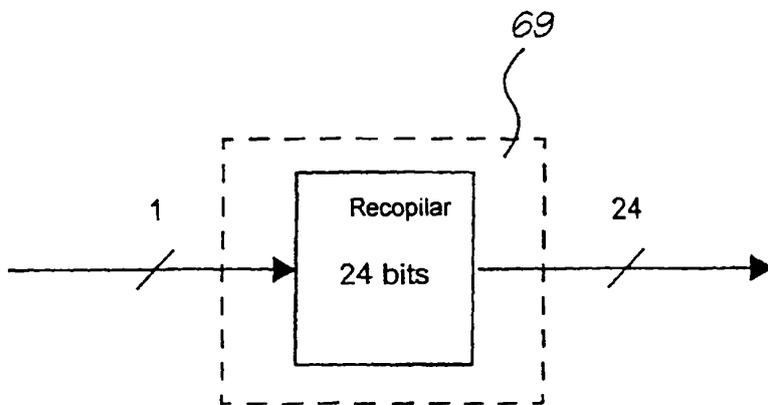


FIG. 32

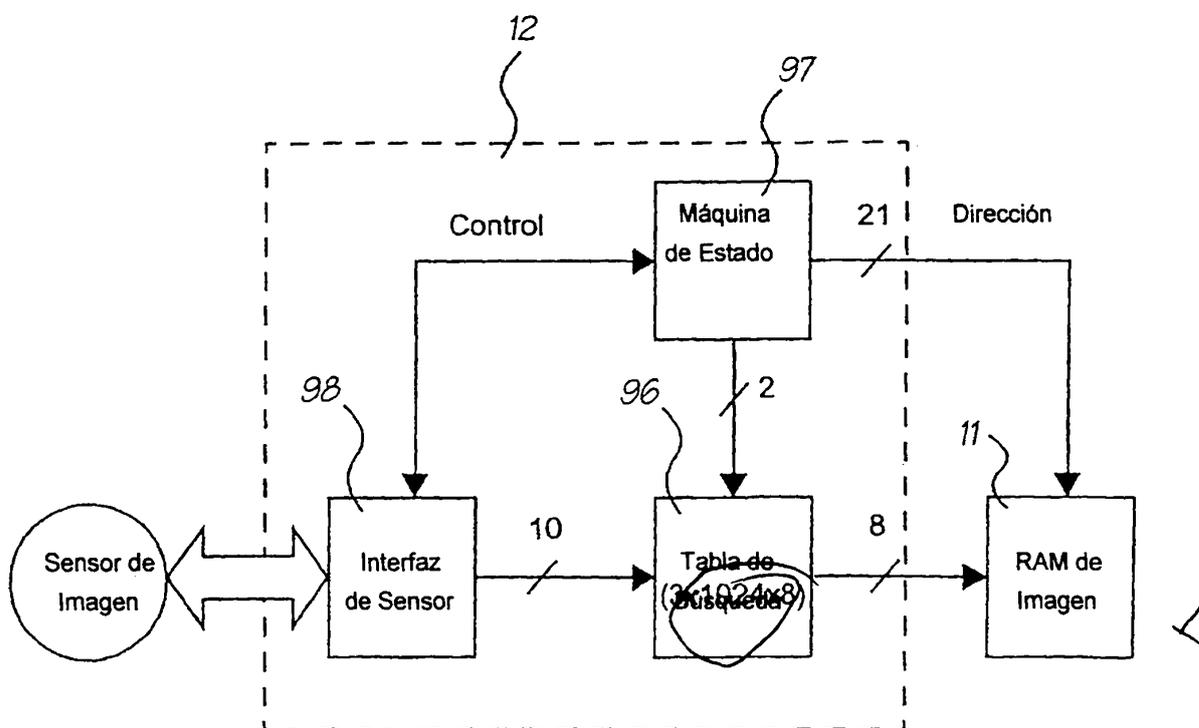


FIG. 33

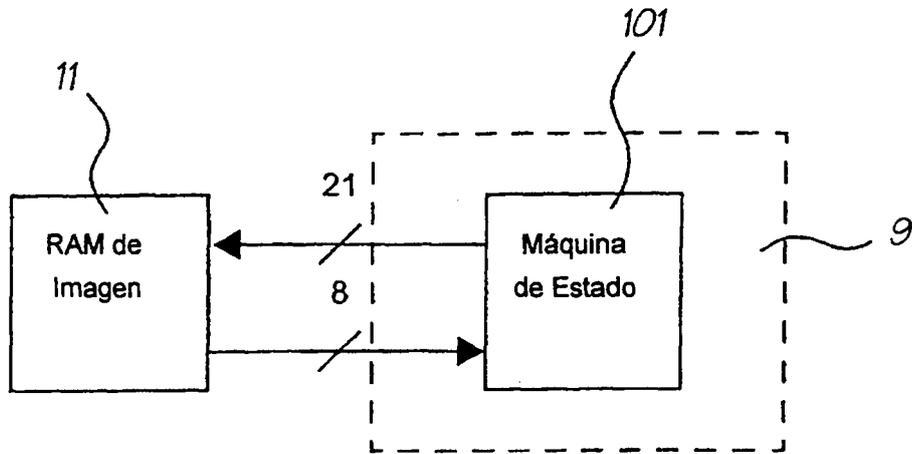


FIG. 35

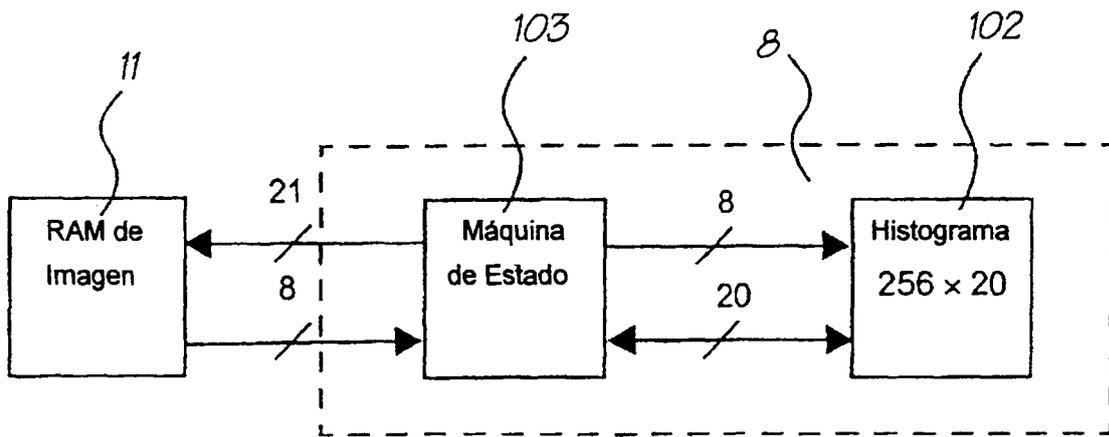


FIG. 36

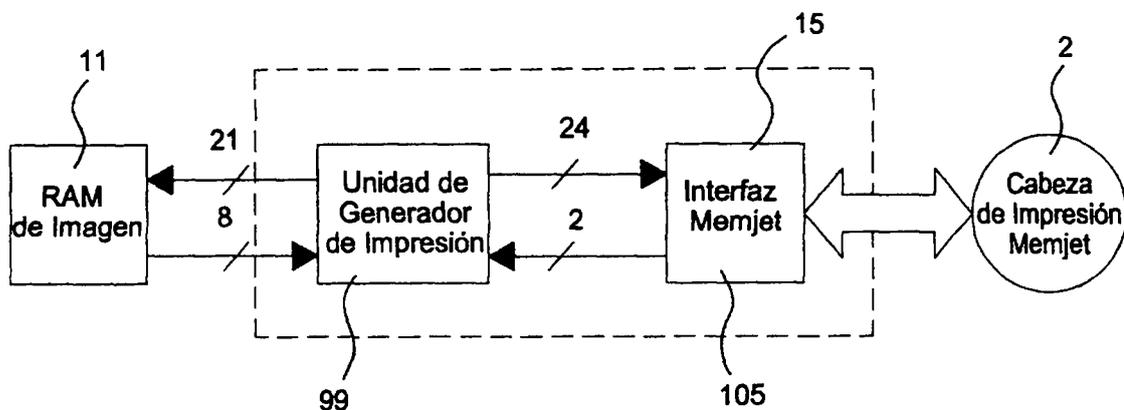


FIG. 37

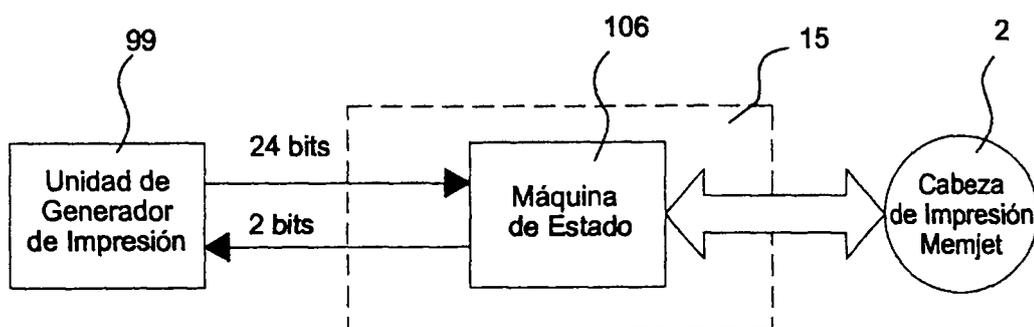


FIG. 38

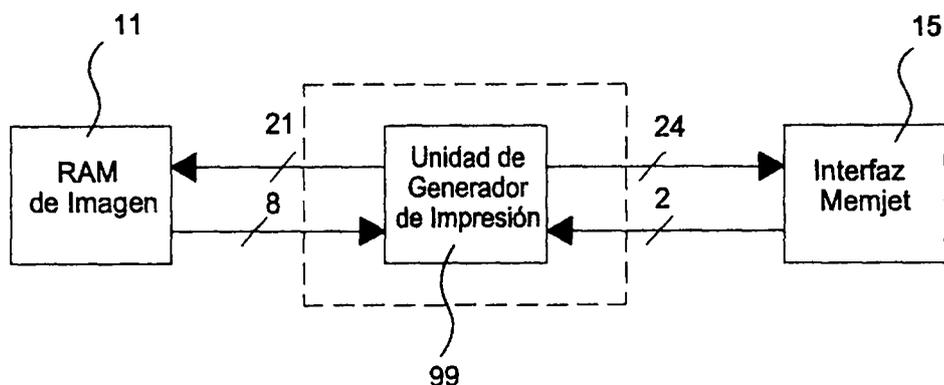


FIG. 41

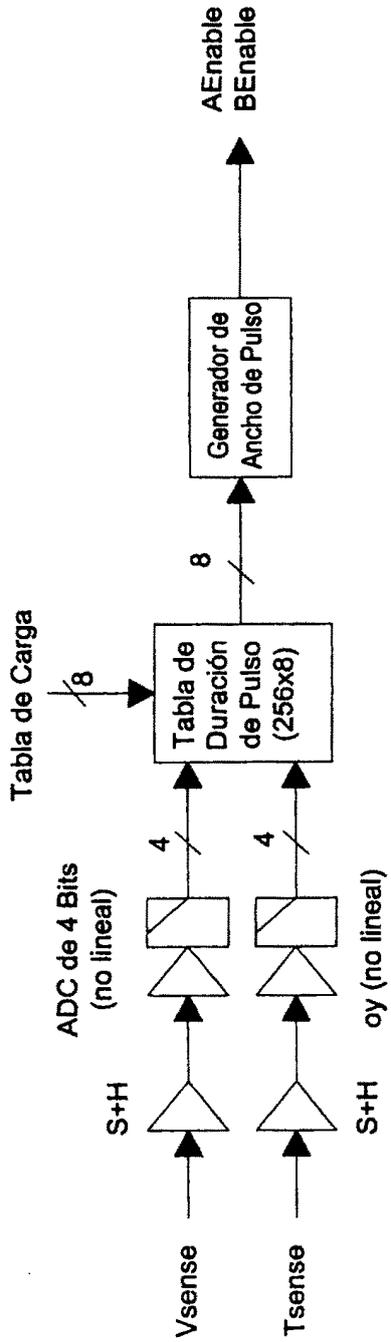


FIG. 39

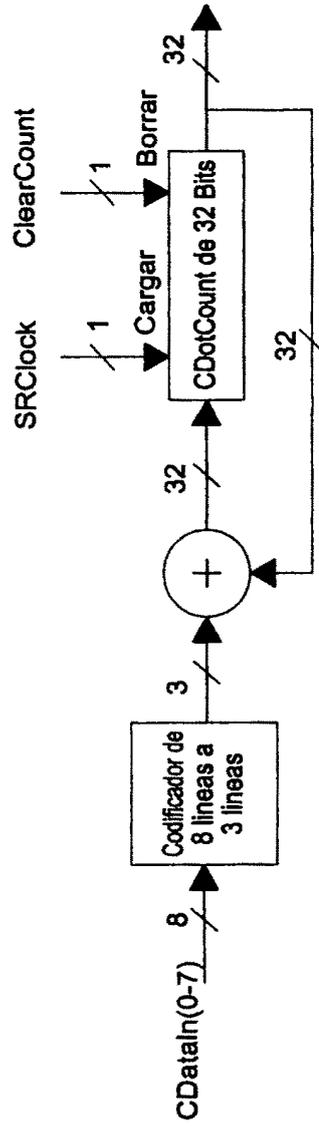


FIG. 40

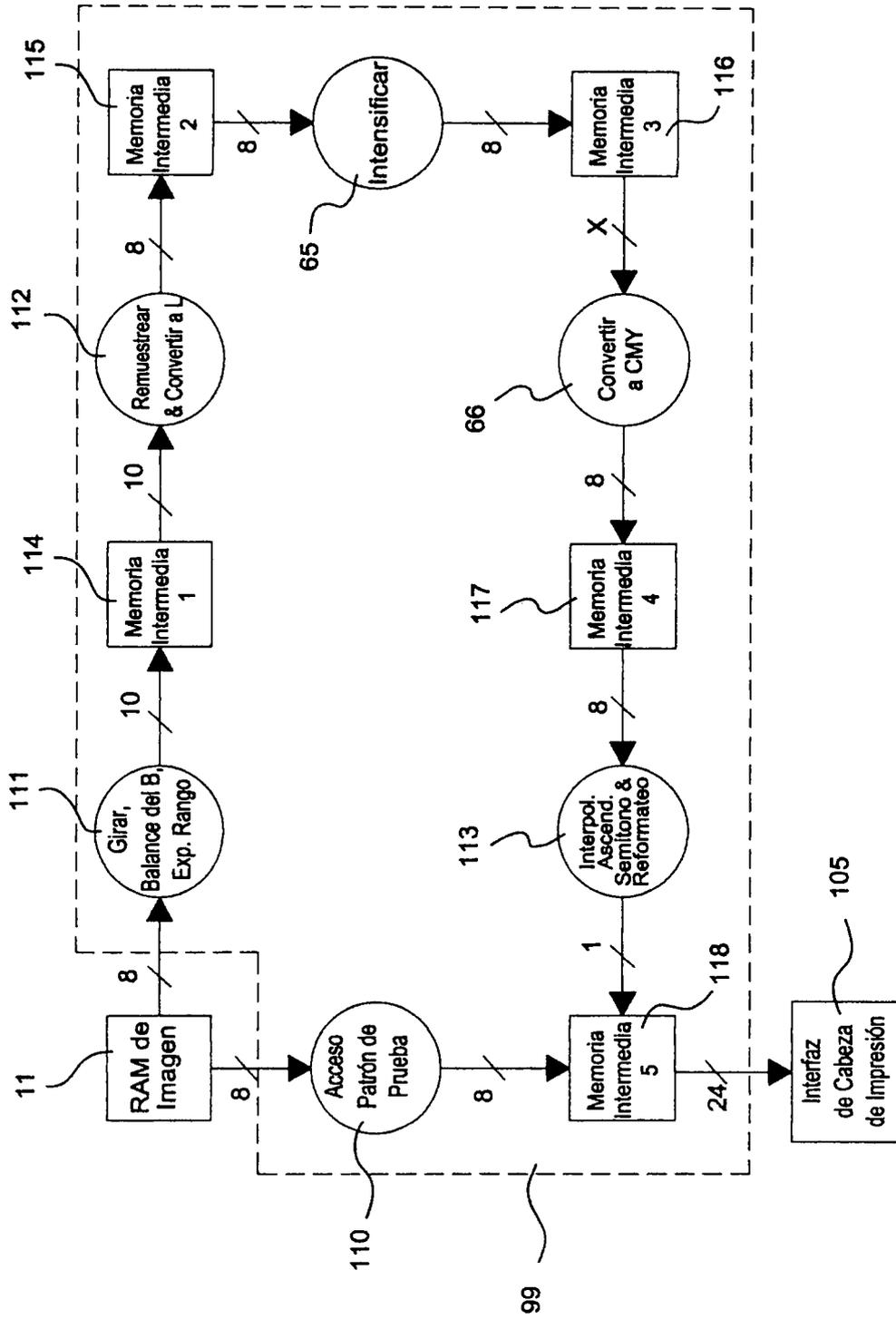


FIG. 42

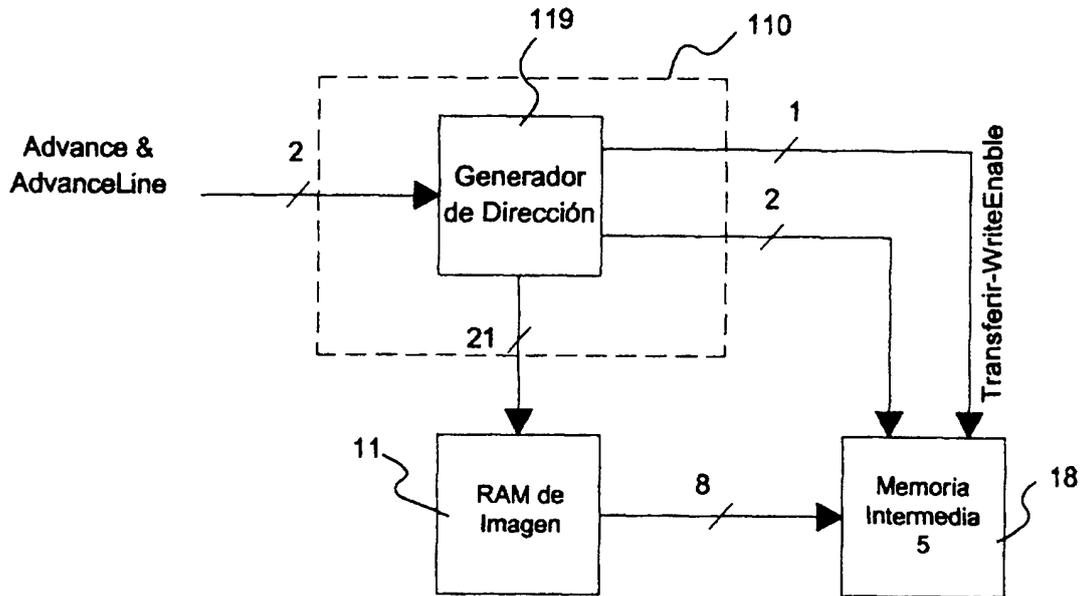


FIG. 43

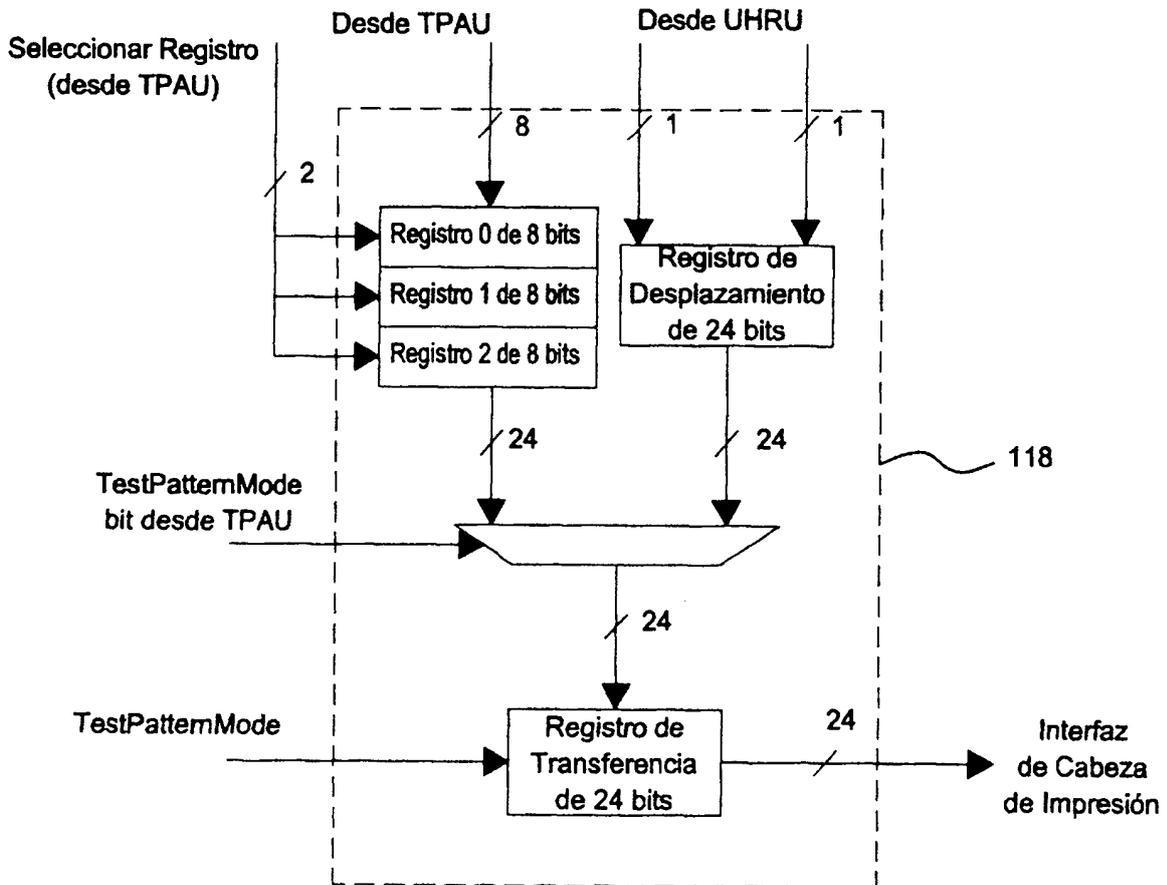


FIG. 44

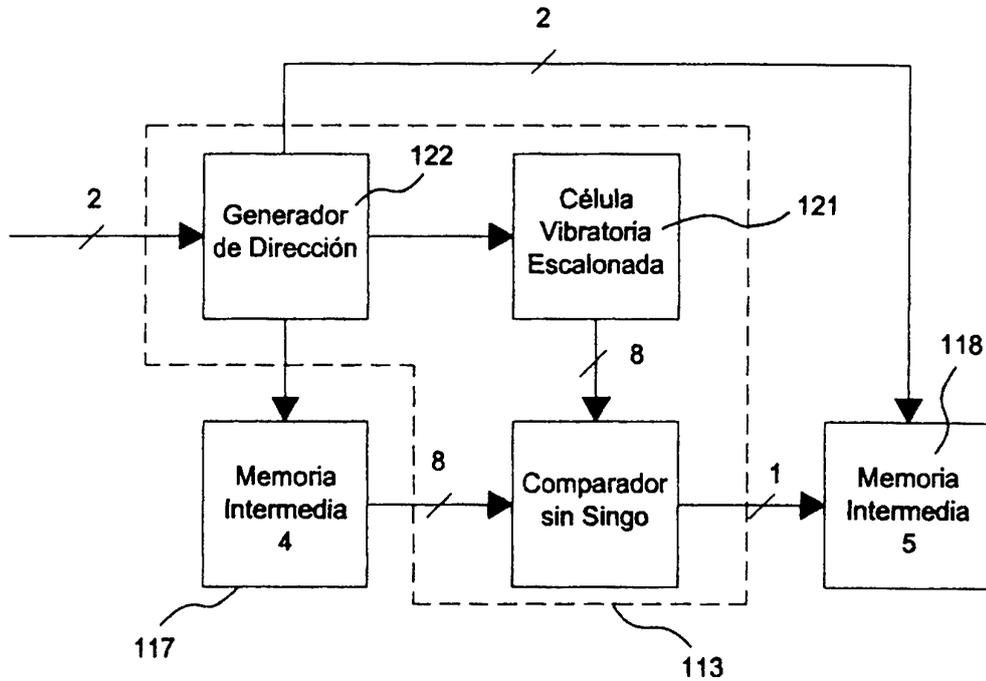


FIG. 46

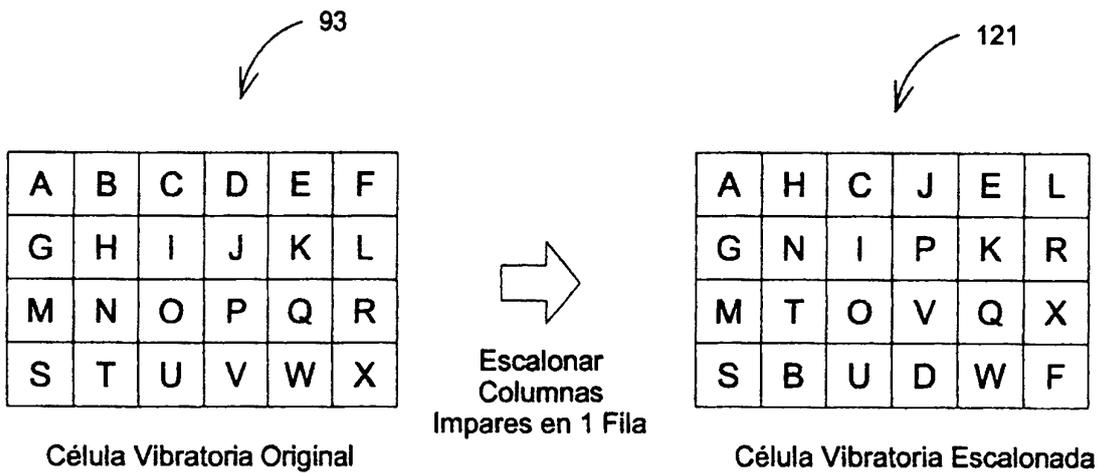


FIG. 47

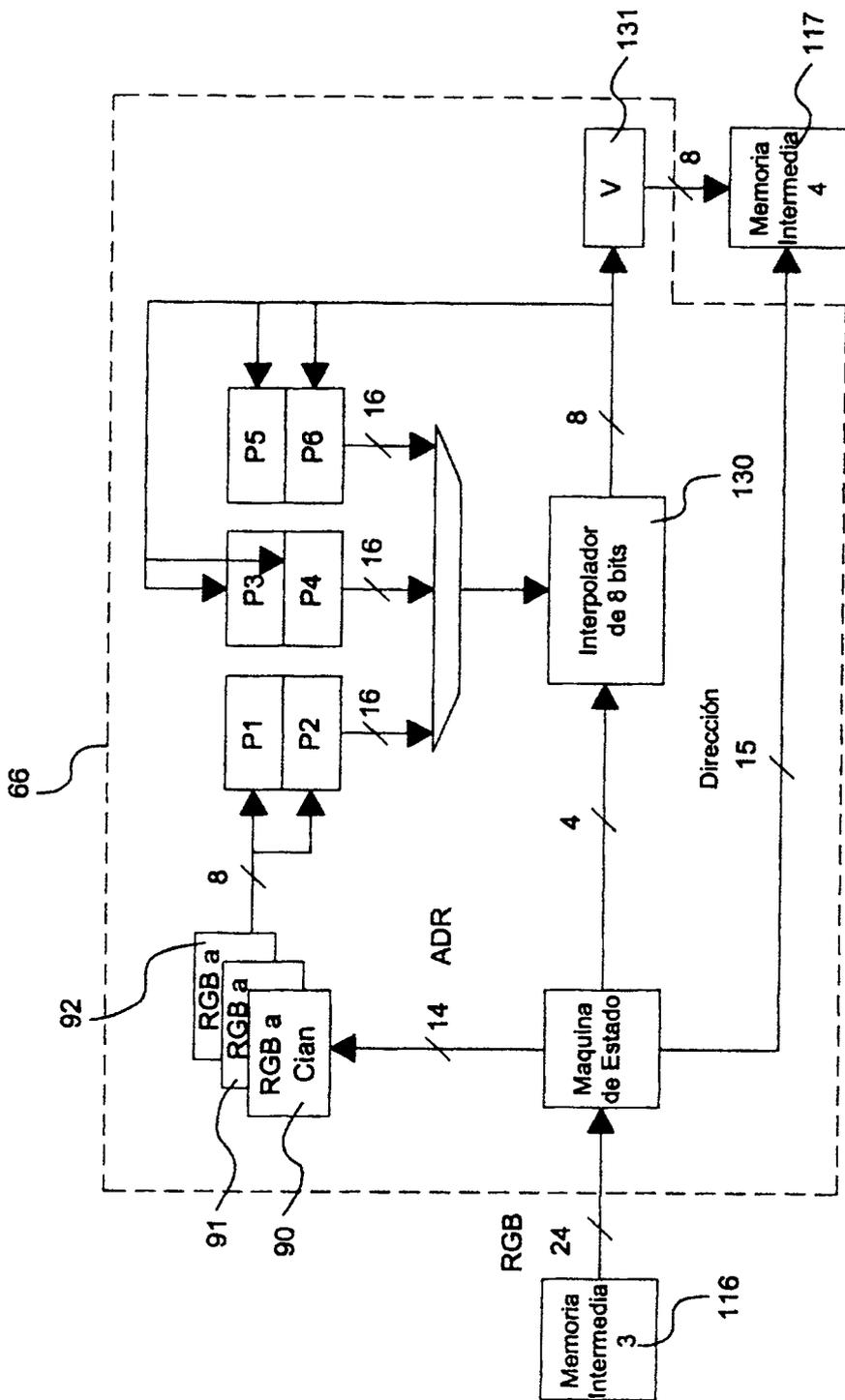


FIG. 48

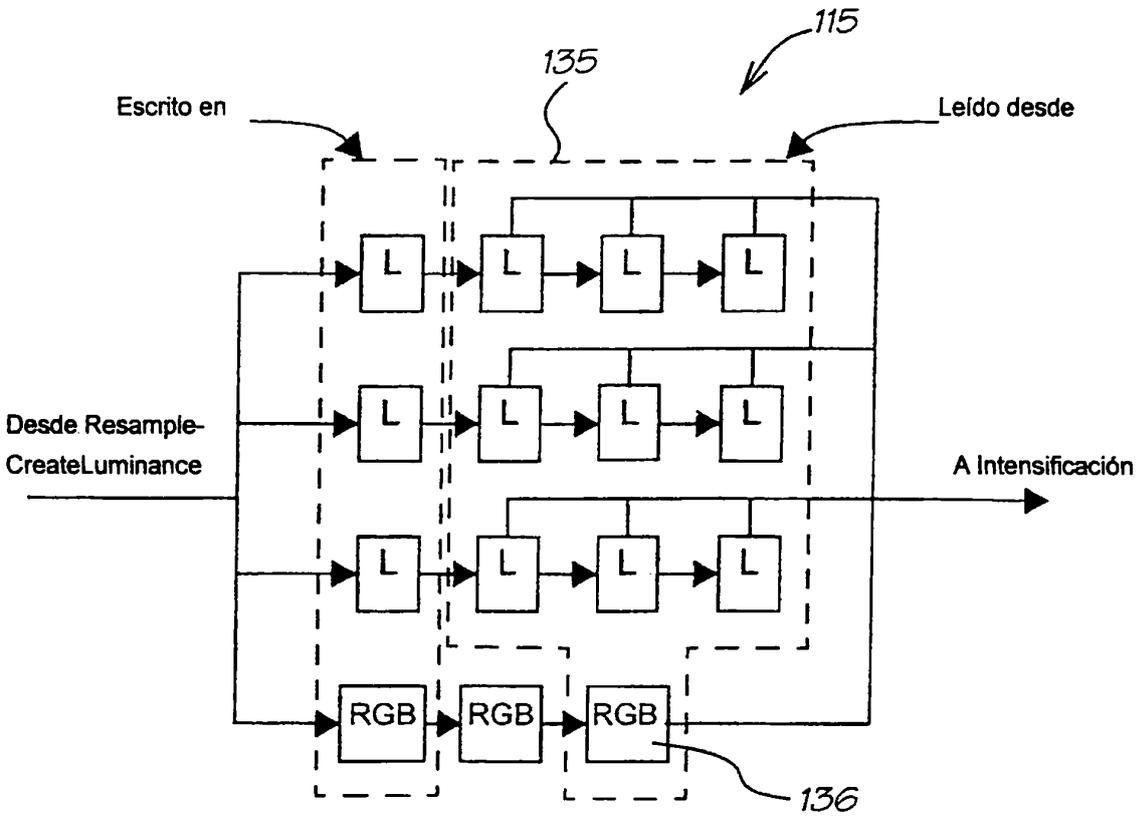


FIG. 49

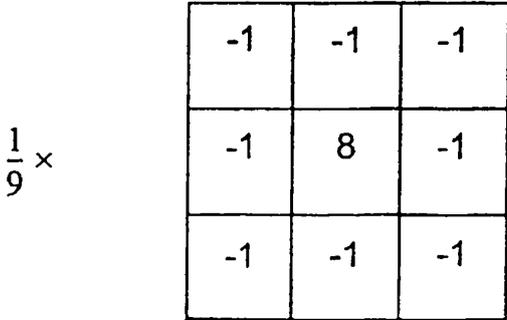


FIG. 50

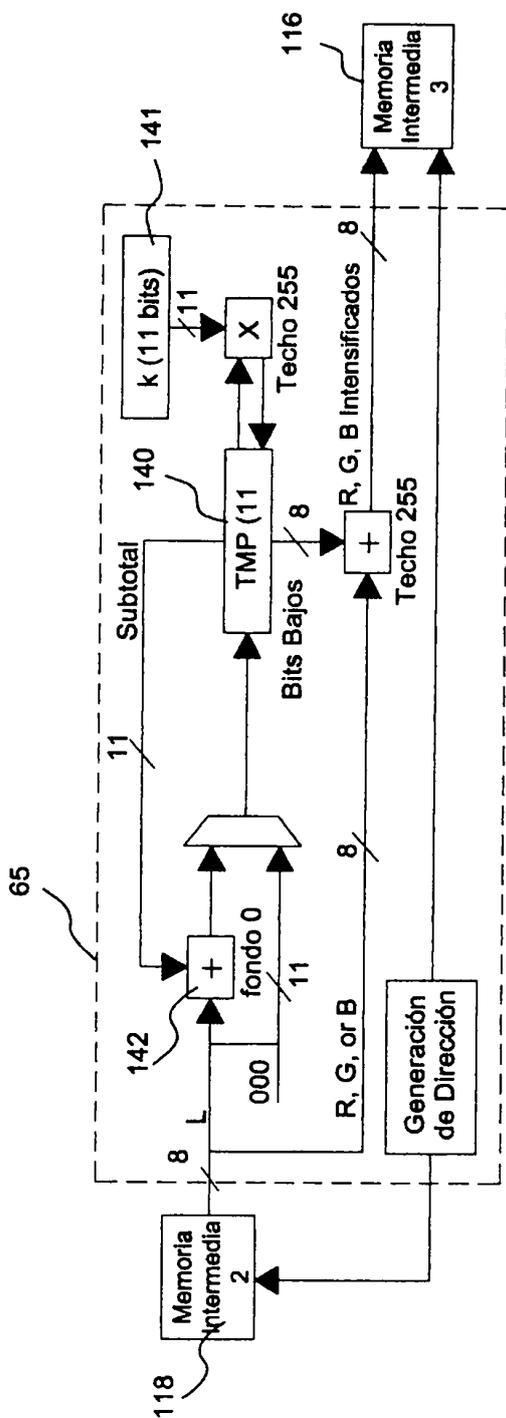


FIG. 51

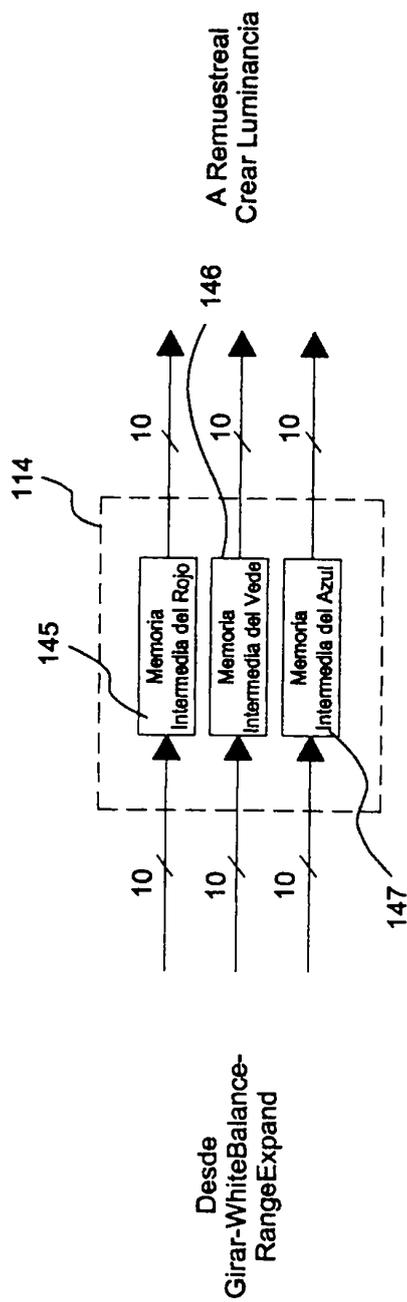


FIG. 52

Desde
Girar-WhiteBalance-
RangeExpand

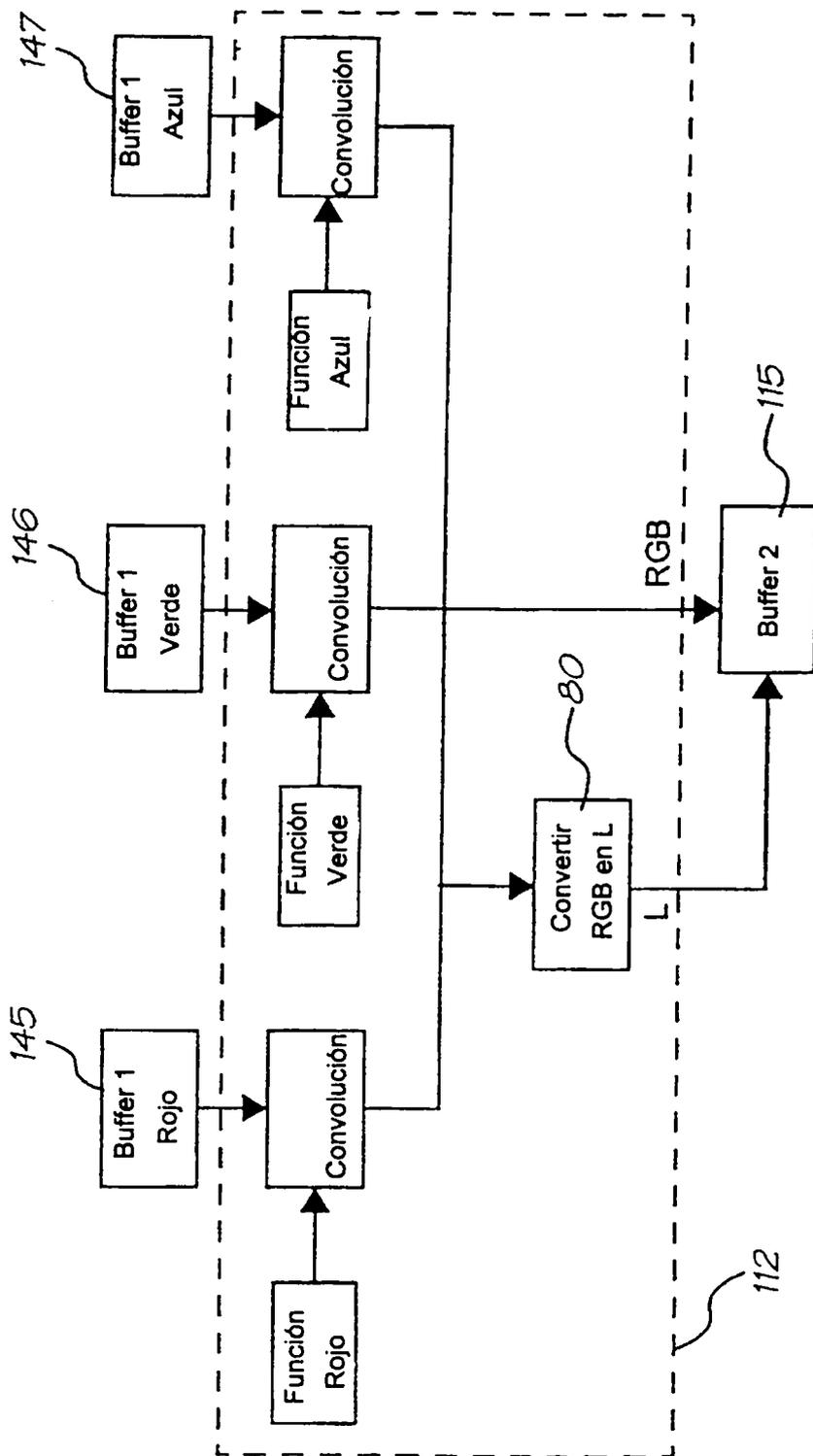


FIG. 53

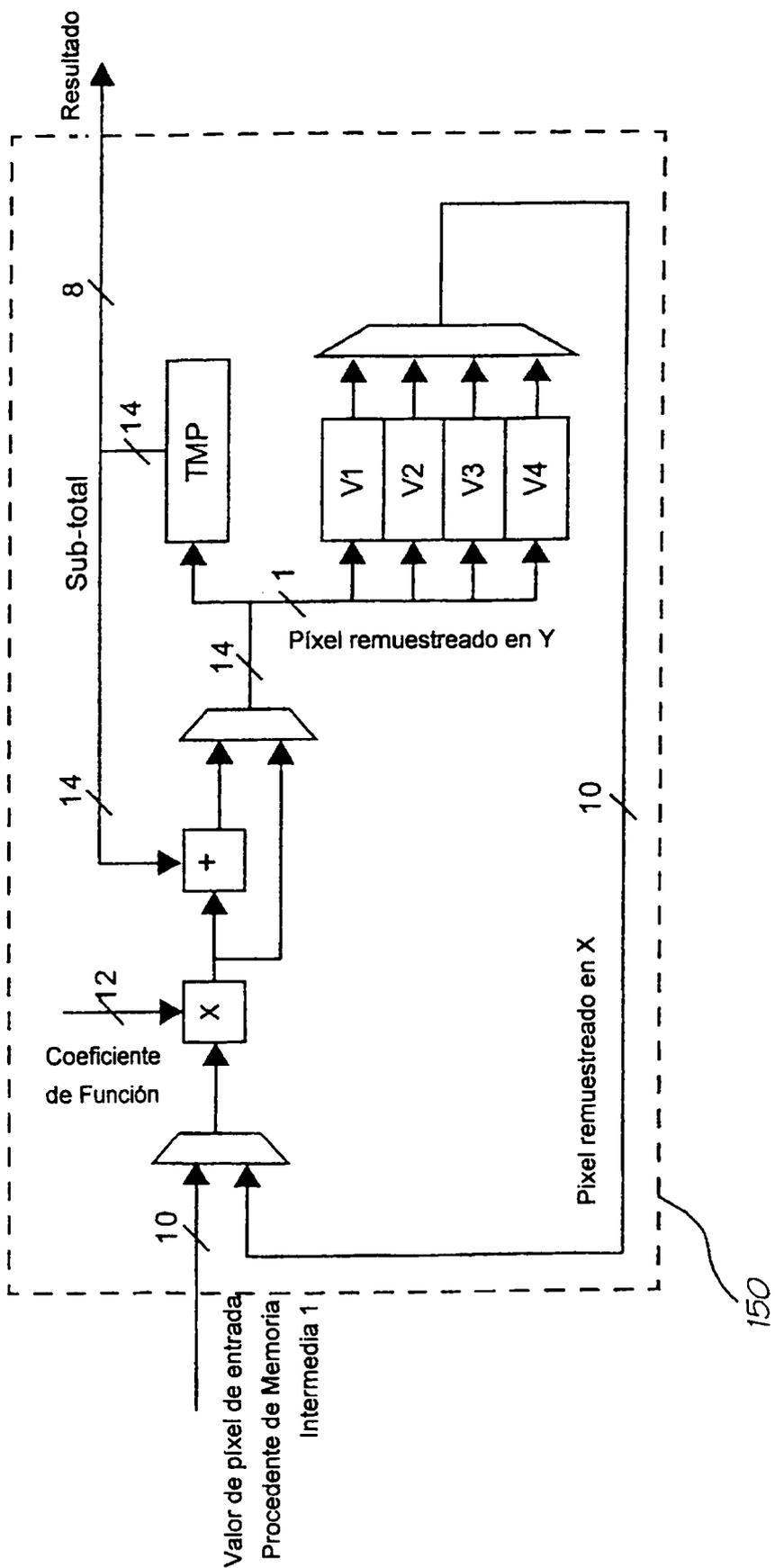


FIG. 54

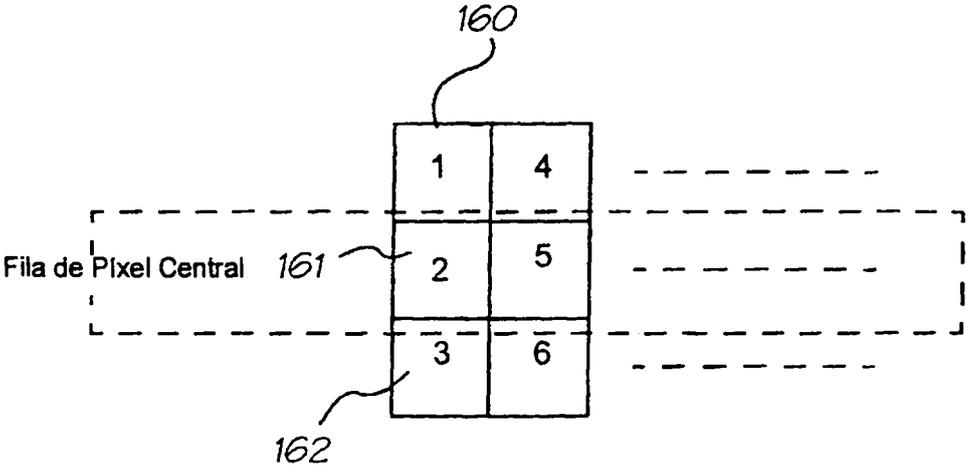
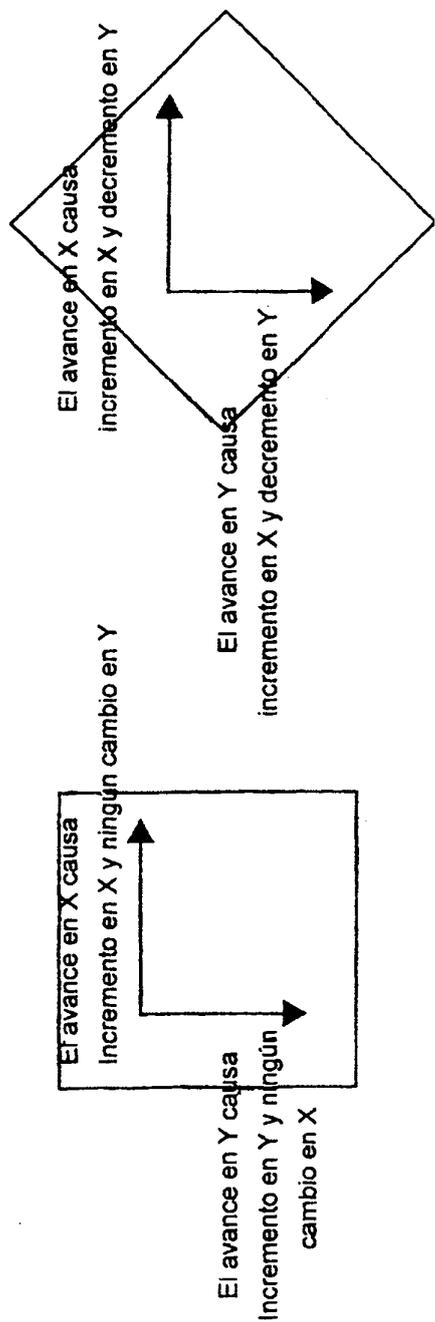


FIG. 55

0	13	26	39	52	65
1	14	27	40	53	66
10	23	36	49	62	75
11	24	37	50	63	76
12	25	38	51	64	77

FIG. 57



No girado

Girado

FIG. 56

	G
G	

Filas pares tienen muestras en píxeles impares
Filas impares tienen muestras en píxeles pares

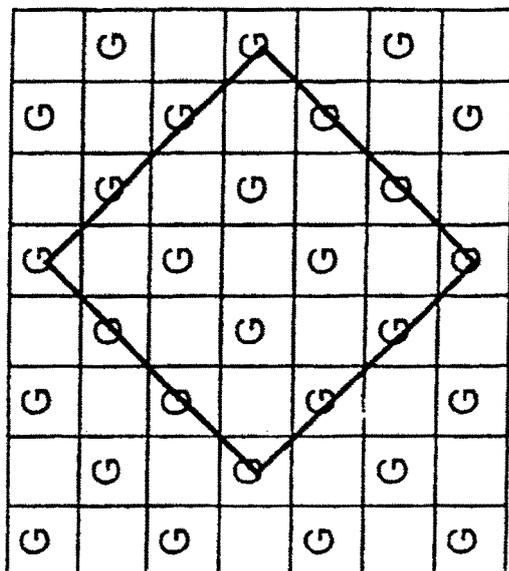
G	
	G

Filas pares tienen muestras en píxeles pares
Filas impares tienen muestras en píxeles impares

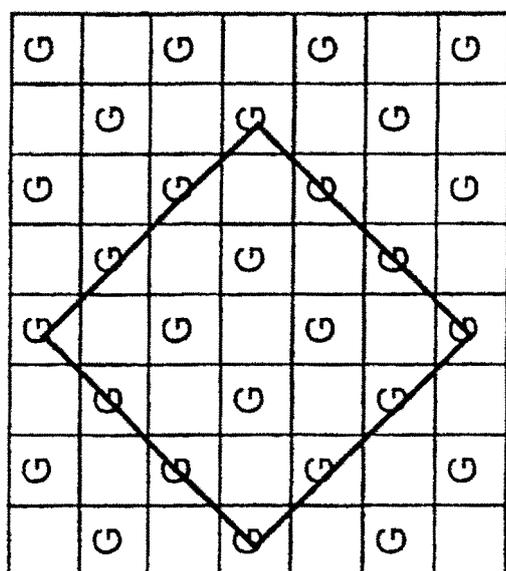
Célula CFA 2 x 2 básica (rotación = 0)

Célula CFA 2 x 2 Básica (rotación = 90)

FIG. 58



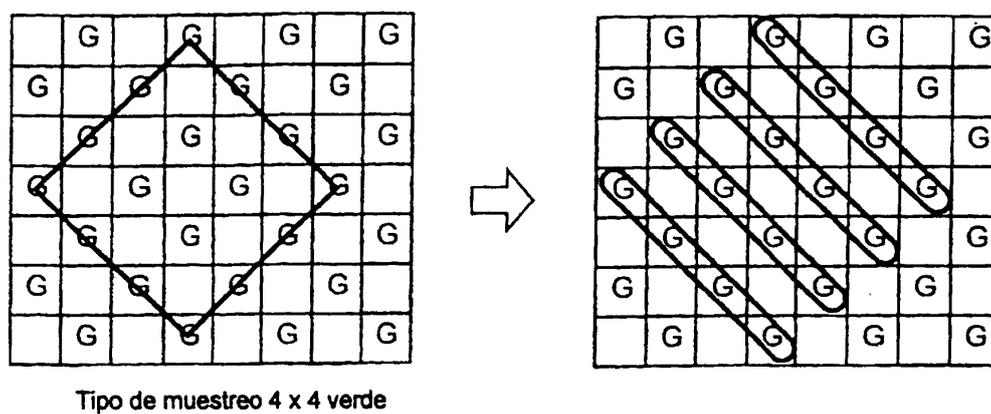
Tipo 2 de muestreo 4 x 4 verde



Tipo 1 de muestreo 4 x 4 verde

Fila impar (3)

FIG. 59



Tipo de muestreo 4 x 4 verde

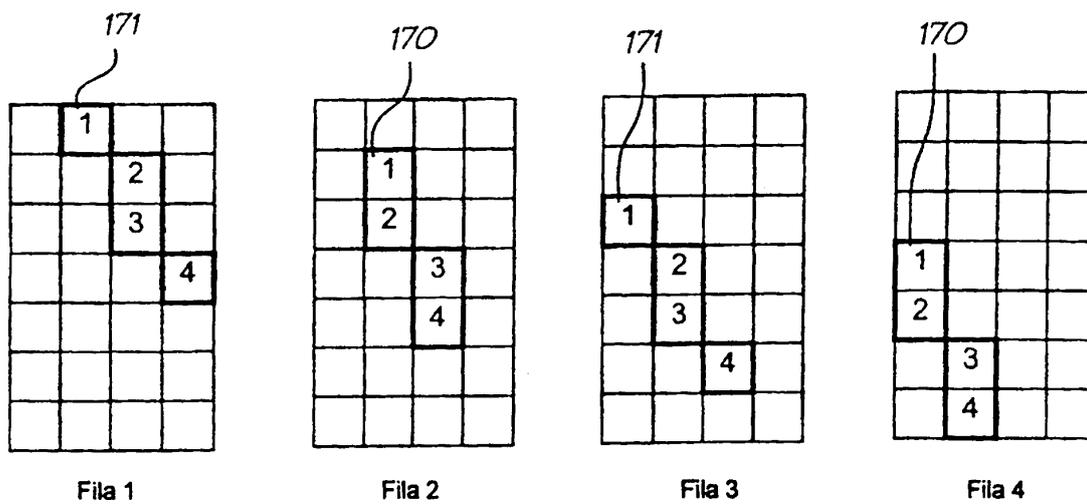
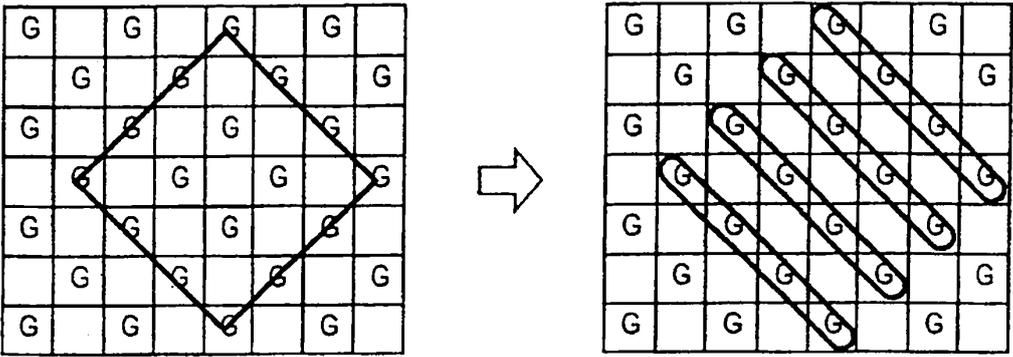


FIG. 60



Tipo 2 de muestreo 4 x 4 verde

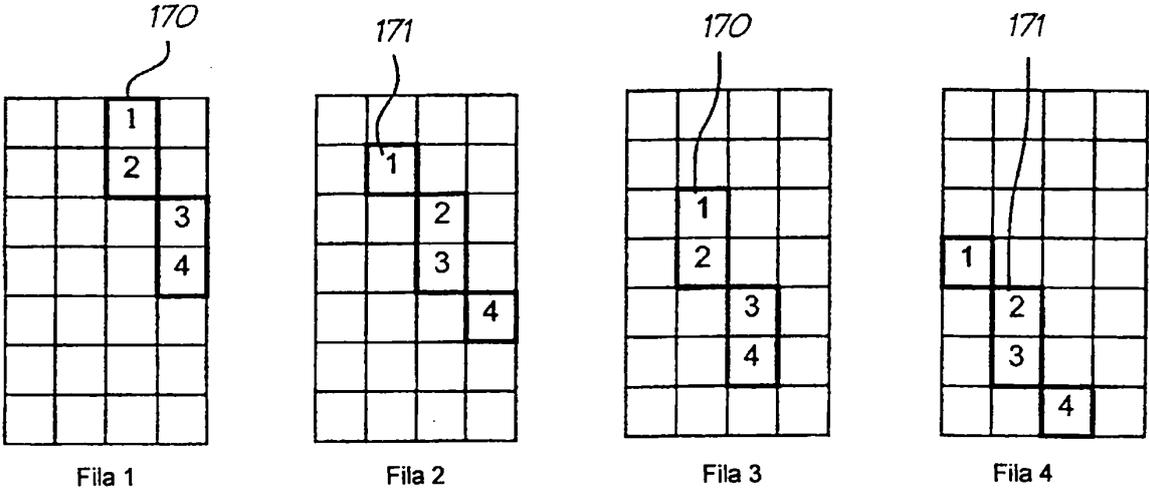


FIG. 61

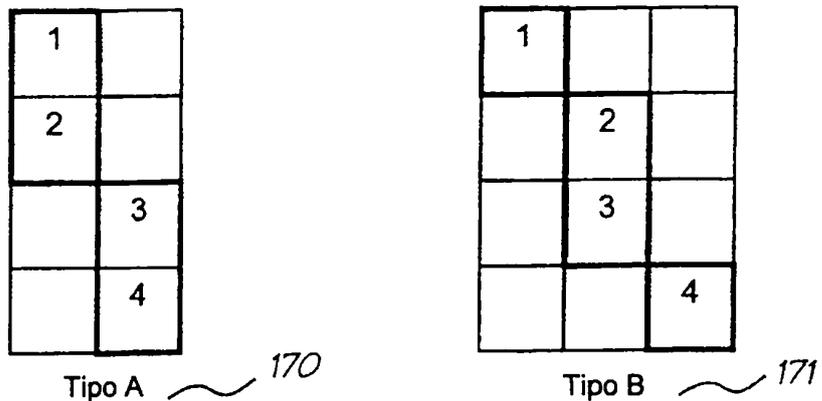


FIG. 62

0	6	12	18	24	30
1	7	13	19	25	31
2	8	14	20	26	32
3	9	15	21	27	33
4	10	16	22	28	34
5	11	17	23	29	35

FIG. 63

1	2	3	4		
5	6	7	8		
9	10	11	12		
13	14	15	16		

FIG. 64

1	2	3	4	
5	6	7	8	
9	10	11	12	
13	14	15	16	

16 muestras leídas para el 3° Pixel

1	2	3	4	
5	6	7	8	
9	10	11	12	
13	14	15	16	

16 muestras leídas para el 2° Pixel

1	2	3	4	
5	6	7	8	
9	10	11	12	
13	14	15	16	

16 muestras leídas para el 1° Pixel

FIG. 65

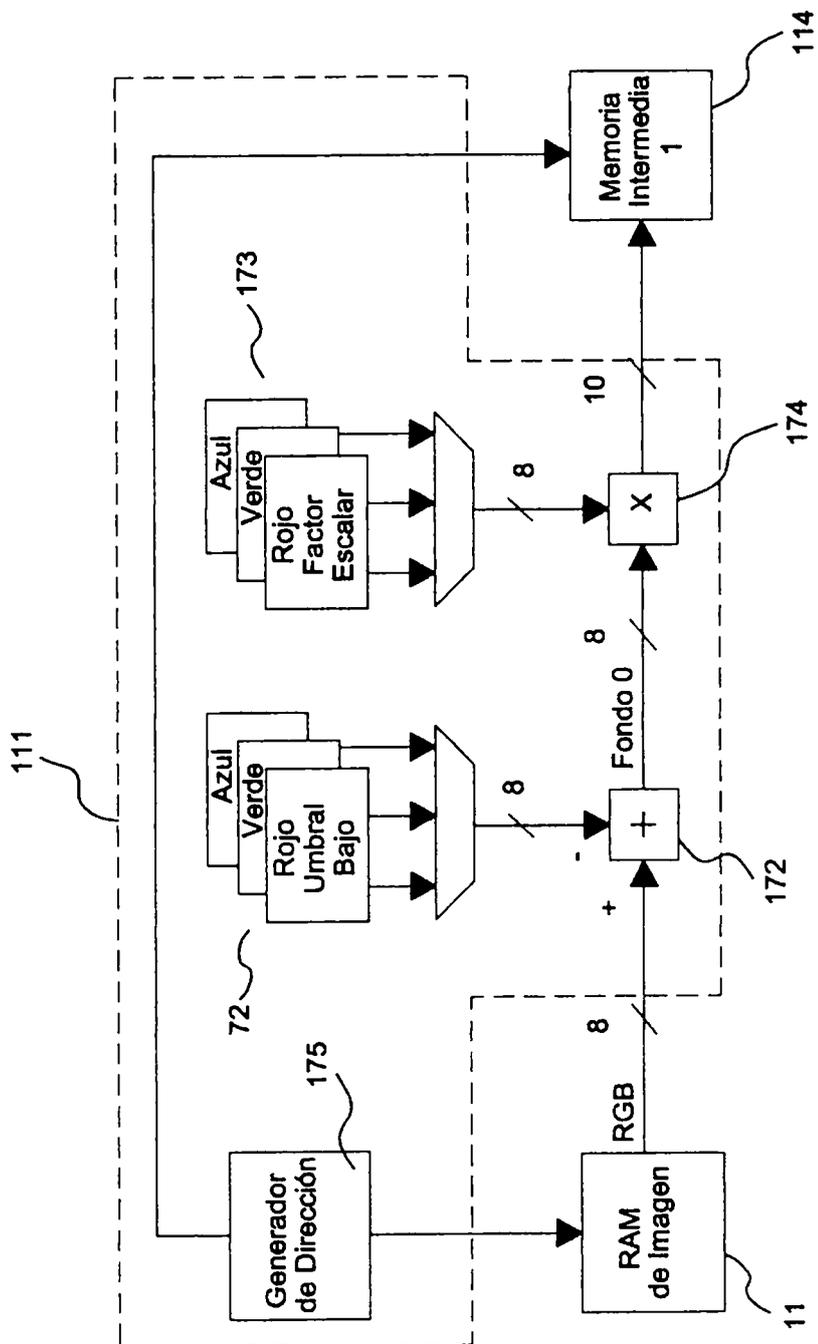


FIG. 66

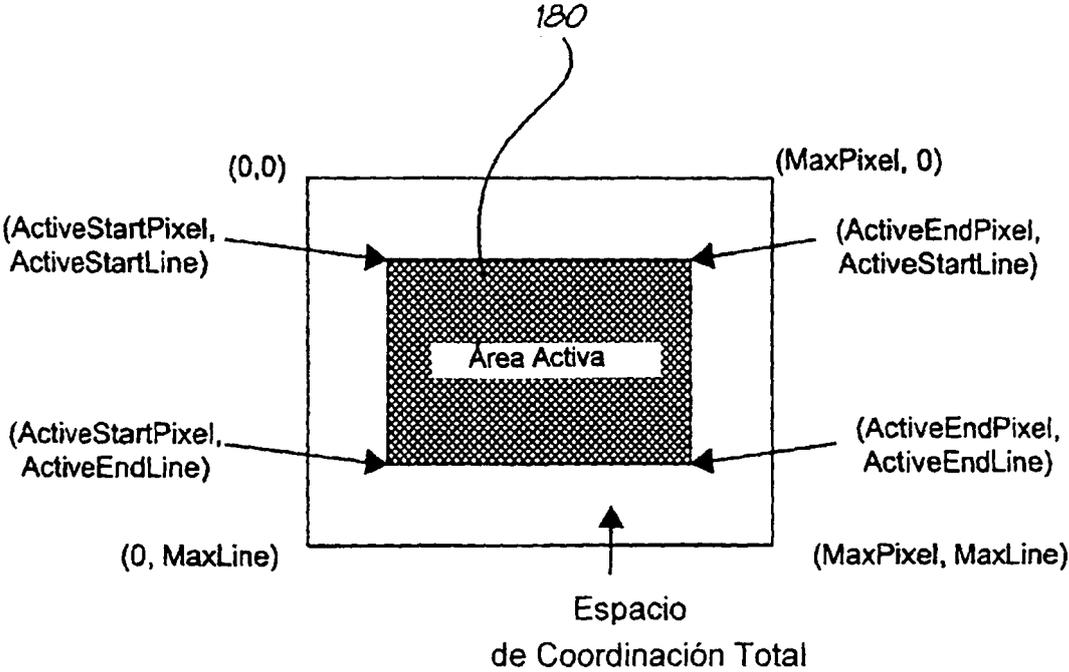


FIG. 67