

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 964 763**

51 Int. Cl.:

H04N 19/11 (2014.01)

H04N 19/14 (2014.01)

H04N 19/18 (2014.01)

H04N 19/176 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **30.03.2020** **E 20166692 (2)**

97 Fecha y número de publicación de la concesión europea: **04.10.2023** **EP 3754983**

54 Título: **Decisión de intra-codificación temprana**

30 Prioridad:

20.06.2019 NO 20190764

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

09.04.2024

73 Titular/es:

**PEXIP AS (100.0%)
Lilleakerveien 2A
0283 Oslo, NO**

72 Inventor/es:

ENDRESEN, LARS PETTER

74 Agente/Representante:

ROEB DÍAZ-ÁLVAREZ, María

ES 2 964 763 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

Decisión de intra-codificación temprana

5 **Campo técnico**

La presente divulgación se refiere a codificación/decodificación intra-bloque en sistemas de compresión de vídeo en dispositivos o sistemas informáticos.

10 **Antecedentes**

La transmisión de imágenes en movimiento en tiempo real se emplea en varias aplicaciones tales como, aunque sin limitación, videoconferencias, reuniones en red, emisiones de televisión (TV) y videotelefonía. La representación de imágenes en movimiento requiere gran cantidad de información ya que el vídeo digital suele describirse representando cada píxel en una imagen con 8 bits, lo que equivale a 1 byte. Tales datos de vídeo sin comprimir dan como resultado grandes volúmenes de bits y no pueden transferirse en tiempo real a través de redes de comunicación y líneas de transmisión convencionales debido al ancho de banda limitado.

Por tanto, habilitar la transmisión de vídeo en tiempo real requiere un alto grado de compresión de datos. Sin embargo, la compresión de datos puede comprometer la calidad de la imagen. Por ello, se han hecho grandes esfuerzos para desarrollar técnicas de compresión que permitan la transmisión en tiempo real de vídeo de alta calidad a través de conexiones de datos con un ancho de banda limitado. En los sistemas de compresión de vídeo, el objetivo principal es representar la información de vídeo con la menor capacidad posible. La capacidad se define en bits, ya sea como un valor constante o como bits/unidad de tiempo. En ambos casos, el objetivo es reducir el número de bits. Dado que los dispositivos usados para la compresión de vídeo son cada vez más pequeños, también es un objetivo reducir la capacidad de procesador requerida. Un método convencional de codificación de vídeo se describe en las normas del Grupo de Expertos en Imágenes en Movimiento (MPEG) y H.26*, VPX y AV1. Los datos de vídeo experimentan cuatro procesos principales antes de su transmisión (es decir, el proceso de predicción, el proceso de transformación, el proceso de cuantificación y la codificación de entropía).

El proceso de predicción reduce el número de bits requerido para cada imagen en una secuencia de vídeo que se va a transferir. El proceso aprovecha la similitud de partes de la secuencia con otras partes de la secuencia. Dado que la parte del predictor es conocida tanto por el codificador como por el decodificador, solo se debe transferir la diferencia. Esta diferencia habitualmente requiere mucha menos capacidad para su representación. La predicción se basa principalmente en vectores que representan movimientos. El proceso de predicción convencionalmente se realiza en tamaños de bloque cuadrados (p. ej., 16x16 píxeles o 64x64 píxeles). En algunos casos, se usan predicciones de píxeles basándose en píxeles adyacentes en la misma instantánea, en lugar de píxeles de instantáneas precedentes. Esto se denomina intra-predicción o codificación intra-fotograma. La codificación intra-fotograma se usa en la codificación de vídeo (compresión).

La predicción intra-fotograma aprovecha la redundancia espacial, es decir, la correlación entre píxeles dentro de un fotograma, calculando valores de predicción por medio de la extrapolación a partir de píxeles ya codificados para una codificación delta eficaz. Es una de las dos clases de métodos de codificación predictiva en codificación de vídeo. Su contrapartida es la predicción inter-fotograma que aprovecha la redundancia temporal. Los denominados intra-fotogramas codificados temporalmente de manera independiente usan únicamente intra-codificación. Los fotogramas predichos, codificados temporalmente, (p. ej., los fotogramas P y B de MPEG) pueden usar predicción intra e inter-fotograma.

Normalmente, las muestras (o bloques) adyacentes conocidas están arriba, arriba a la izquierda, arriba a la derecha y a la izquierda.

Normalmente, solo se usan unas pocas de las muestras conocidas espacialmente más cercanas para la extrapolación. Los formatos basados en bloques (transformada de frecuencia) rellenan previamente bloques completos con valores de predicción normalmente extrapolados a partir de una o dos líneas rectas de píxeles que discurren a lo largo de sus bordes superior e izquierdo.

La expresión codificación intra-fotograma se refiere al hecho de que las diversas técnicas de compresión sin pérdidas y con pérdidas se realizan con relación a información que está contenida únicamente dentro del fotograma actual, y no con relación a cualquier otro fotograma de la secuencia de vídeo. En otras palabras, no se realiza ningún procesamiento temporal fuera de la instantánea o fotograma actual.

El residual representado como un bloque de datos todavía contiene una correlación interna. Un método convencional que aprovecha esto y realiza una transformada de bloque bidimensional. En algunas normas se usa una transformada de tipo entero, y en otras normas se usa una transformada de coseno discreta (DCT). La dimensión (N*N) de los bloques puede variar, pero, p. ej. se pueden usar transformadas de DCT VP9, 4x4, 8x8, 16x16 y 32x32. Esto transforma N*N píxeles en N*N coeficientes de transformada que normalmente pueden representarse con menos bits

que la representación de píxeles. La transformada de una matriz de $N \times N$ de píxeles con correlación interna puede dar como resultado un bloque de $N \times N$ de coeficientes de transformada con muchos menos valores distintos de cero que el bloque original de $N \times N$ píxeles.

- 5 La codificación de vídeo para formatos HD aumenta las demandas de memoria y procesamiento de datos y requiere organizaciones de memoria eficientes y de alto ancho de banda junto con capacidades de cálculo intensivo. Debido a estas múltiples demandas, se debe encontrar un enfoque flexible de procesamiento paralelo para satisfacer las demandas de una manera rentable.
- 10 La codificación intra-fotograma es normalmente una técnica de codificación más costosa en términos de procesamiento de datos que la codificación inter-fotograma. Sin embargo, en algunos casos, la inter-fotograma puede tener un impacto negativo en la calidad visual de la imagen vídeo decodificada. Por ejemplo, en regiones uniformes de la instantánea de vídeo que todavía no tienen el mismo color y/o brillo de manera consistente por toda la región, un espectador las percibirá como "planas" y poco naturales. Lo mismo ocurriría con el fondo de los objetos en
- 15 movimiento, donde los espectadores podrían ver de hecho restos de fotogramas de los bloques de inter-codificación después de que haya pasado un objeto, artefactos de seguimiento, que se percibirían como ruido visual.

Por lo tanto, existe la necesidad de conseguir un método para detectar estos casos y usar intra-codificación para evitar ruido visual de la inter-codificación sin aumentar significativamente el consumo de capacidad de procesador.

- 20 El documento de Gang He y col.: "Transformed-based fast mode and depth decision algorithm for HEVC intra prediction", 2013 IEEE 10th International Conference On ASIC, IEEE, del 28 de octubre de 2013, páginas 1 - 4, divulga un criterio para seleccionar grandes tamaños de bloque de intra-codificación basándose en la relación de componentes de CC a CA.

- 25 Otros documentos de antecedentes tecnológicos ilustrativos comprenden el documento de Wei Rujun y col.: "Fast depth decision with extended coding block sizes for HEVVC intra coding of 4K ultra-HD video", 2015 IEEE Workshop On Signal Processing Systems (SIPS), IEEE, del 14 de octubre de 2015, páginas 1 - 6; y el documento de Tianruo Zhang y col.: "A Frequency-Based Fast Block Type Decision Algorithm For Intra Predicción In H.264/AVC High Profile", Circuits And Systems, 2008, APCCAS 2008, IEEE Asia Pacific Conference On, IEEE, Piscataway, NJ, EE. UU., del 30 de noviembre de 2008, páginas 1292 - 1295.

Sumario

- 35 En vista de lo anterior, un objetivo de la presente divulgación consiste en superar o al menos mitigar los inconvenientes de los sistemas de videoconferencia de la técnica anterior.

De acuerdo con la invención, se proporciona un método de codificación de vídeo como se expone en la reivindicación 1, y se proporciona un dispositivo procesador de vídeo como se expone en la reivindicación 7.

- 40 Todas las realizaciones divulgadas en la descripción están cubiertas por las reivindicaciones.

Breve descripción de los dibujos

- 45 La Figura 1 ilustra los píxeles en un bloque de cámara de 64×64 , dividido en subbloques de 8×8

La figura 2a-2d ilustra el desarrollo del conjunto de instrucciones de 64 bits de MMX,

La figura 3a es una imagen de vídeo codificada/decodificada

- 50 La figura 3b es la imagen de vídeo codificada/decodificada de la figura 3a con bloques intra-codificados marcados en gris claro.

Descripción detallada

- 55 De acuerdo con las realizaciones de la presente invención, como se divulga en el presente documento, las desventajas mencionadas anteriormente de las soluciones de acuerdo con la técnica anterior se eliminan o al menos se mitigan.

- 60 La Figura 1 ilustra un bloque de cámara de $M \times M$ dividido en $N \times N$ bloques, que es un bloque de cámara de 64×64 dividido en 64 bloques de 8×8 , lo que significa 64 píxeles en cada bloque de 8×8 . En la siguiente descripción, se hará referencia a esta división/agrupación de píxeles, que cumple con la norma VP9, AV1 y H.265 mencionada anteriormente.

- 65 La idea básica de la presente invención es usar grandes intra-bloques para regiones uniformes del vídeo que toman una decisión temprana de codificación de intra-bloque basándose en cálculos de DCT y CC. Esto aumentará la calidad visual a un coste moderado ya que pueden evitarse investigaciones adicionales, decisión de modo, búsqueda de

movimientos y así sucesivamente usando un cálculo paralelo.

En determinadas realizaciones, de acuerdo con la presente solicitud, esto se consigue mediante una primera etapa que calcula la DCT de cada bloque de 8x8 y luego posiblemente usando el valor de CC para calcular el residual. Para averiguar si esa predicción de CC es una buena elección para el residual del bloque de 8x8, los coeficientes de la DCT correspondiente se comparan con un primer umbral predefinido. Cuando todos estos coeficientes están por debajo de un umbral determinado, se considera que esa predicción de CC es una buena elección para el bloque de 8x8. Será evidente en la siguiente exposición, que la predicción de CC como una buena elección significaría seleccionar intra-codificación.

El inventor se ha dado cuenta de que esto es una buena coincidencia con el conjunto de instrucciones vectoriales Intel MMX de 64 bits. MMX es un conjunto de instrucciones de instrucción única y datos múltiples (SIMD) diseñado por Intel e introducido en 1997 con su línea de microprocesadores Pentium basados en P5, denominada "Pentium con tecnología MMX". Se desarrolló a partir de una unidad similar introducida en el Intel i860 y, anteriormente, en el procesador de píxeles de vídeo Intel i750. MMX es una capacidad complementaria de procesador que está soportada en procesadores IA-32 recientes de Intel y otros proveedores. MMX define ocho registros, denominados MM0 a MM7, y operaciones que operan en los mismos. Cada registro tiene 64 bits de ancho y se puede usar para contener ya sea números enteros de 64 bits o múltiples números enteros más pequeños en un formato "empaquetado": una única instrucción se puede aplicar a dos números enteros de 32 bits, cuatro números enteros de 16 bits u ocho números enteros de 8 bits a la vez.

Cabe señalar que, a partir de ahora, el conjunto de MMX se denominará conjunto de instrucciones vectoriales de Intel MMX de 64 bits o cualquier otro conjunto de instrucciones vectoriales con características y rendimiento similares.

El conjunto de instrucciones AVX512 de 64 bits permite calcular el valor de DCT de 8 bloques de 8x8 en paralelo usando, p. ej., la instrucción `64bit_mm512_sad_epu8`. El resultado de la DCT de 8x8 se puede almacenar en una primera máscara de bits de 64 bits, puesto que un bloque de 8x8 tiene 64 píxeles y la DCT del bloque de 8x8 tiene 64 coeficientes. En esta máscara, un "1" significa que el coeficiente está por encima del primer umbral predefinido y "0" significa que el coeficiente está por debajo del primer umbral predefinido. Si los 64 coeficientes son todos cero, entonces la máscara de bits de 64 bits es cero y la predicción de CC del bloque de 8x8 es una buena elección.

En un bloque de 64x64 hay en total 64 bloques de 8x8. En determinadas realizaciones, para cada bloque de 8x8 se establece un "1" si al menos un coeficiente es distinto de cero y un "0" si todos los coeficientes son cero, lo que da como resultado una segunda máscara de bits de 64 bits en un nivel superior, esta para todos los 64 bloques de 8x8 en el bloque de 64x64. Si los 64 coeficientes son todos cero, entonces la máscara de bits de 64 bits es cero, y la predicción de CC del bloque de 64x64 puede ser una buena elección.

Sin embargo, existe la posibilidad de que uno o varios bloques de 8x8 tengan un color uniforme, pero diferente de los otros bloques de 8x8, p. ej. un patrón de "tablero de ajedrez". En tales casos, la prueba anterior no es satisfactoria y posiblemente da como resultado una imagen "plana" en lugar de un patrón de tablero de ajedrez.

Para compensar esta potencial distorsión, el inventor sugiere calcular una tercera máscara de bits de 64 bits. Esto se hace calculando la DCT de las CC del bloque 64 de 8x8 que ya se está calculando. Esto requiere que los valores de CC de cada bloque de 8x8 del bloque de 64x64 completo se almacenen en un bloque de 8x8, donde cada "píxel" contiene el valor de CC de un bloque de 8x8. A continuación, se calcula la DCT de este bloque de 8x8, después, se compara cada coeficiente en el bloque de 8x8 de "DCT de CC" con un segundo valor umbral predefinido.

Esto genera una tercera máscara de bits de 64 bits, donde "1" significa que el coeficiente está por encima del segundo umbral predefinido, y "0" significa que el coeficiente está por debajo del segundo umbral predefinido.

Cabe señalar que, el inventor se ha dado cuenta de que, en la mayoría de los casos, el segundo umbral predefinido debería ser algo más bajo que el primer umbral predefinido.

Si tanto la segunda como la tercera (e implícitamente la primera) máscara de bits de 64 bits son cero, entonces se considera que la probabilidad de que el contenido de instantánea del bloque de cámara de 64x64 sea uniforme, y no similar a un tablero de ajedrez, es alta. Entonces se satisface el criterio para la codificación intra-bloque de acuerdo con la presente invención y se seleccionaría la codificación intra-bloque.

El proceso adicional de determinar intra/no intra en el bloque 64x64 se realizará a un coste relativamente bajo, la mejora de la instantánea decodificada será considerable, al menos en algunas partes, ya que los artefactos visuales, particularmente en superficies uniformes, se evitan al omitir la inter-codificación 64x64 cuando ésta sea un problema.

La razón por la que el proceso de determinación puede llevarse a cabo con un coste de procesamiento relativamente bajo es, como se ha indicado anteriormente, porque se adapta bien al actual conjunto de instrucciones vectoriales de Intel.

Esto se basó en un conjunto de instrucciones MMX de 64 bits cuando se introdujo en 1997. En este conjunto de instrucciones MMX de 64 bits, la instrucción de Suma de Diferencias Absolutas (SAD) `_mm_sad_pu8` calculó la suma absoluta de 8 píxeles con otros 8 píxeles.

5 Sin embargo, cuando más tarde se introdujo SSE2 de 128 bits, la instrucción SAD `_mm_sad_epu8` devolvía inesperadamente la suma absoluta en fragmentos de 8 píxeles y no en fragmentos de 16 píxeles, como se ilustra en la figura 2a. La razón era que se reutilizó el hardware de MMX subyacente.

10 Mediante la introducción de AVX2 de 256 bits, sucedió lo mismo, es decir, la instrucción SAD `_mm256_sad_epu8` seguía devolviendo la suma absoluta en fragmentos de 8 píxeles, como se ilustra en la figura 2b.

Además, mediante la introducción de AVX512 de 512 bits, de nuevo sucedía lo mismo, la instrucción SAD `_mm512_sad_epu8` devolvía la suma absoluta en fragmentos de 8 píxeles, como se ilustra en la figura 2c.

15 Cuando se calcula la SAD de todos los 64 píxeles, como se ilustra en la figura 2d, esto supone una pequeña desventaja, ya que se necesita recopilar las 8 SAD parciales que conllevan numerosas instrucciones adicionales.

20 Sin embargo, esto se puede usar como una ventaja en la decisión de intra-codificación temprana. Volviendo ahora a la figura 1, la línea superior del bloque de 64x64 incluye 8 líneas superiores respectivas de los 8 bloques de 8x8 de más arriba del bloque de 64x64. Para calcular los coeficientes de CC y DCT de un bloque de 8x8, la operación básica consiste en calcular la suma de los valores de píxel. De acuerdo con una realización de la presente solicitud, cada suma se realiza en paralelo para los 8 bloques de 8x8 en su totalidad del bloque de 64x64. Esto es posible puesto que la instrucción SAD de AVX512 de 512 bits devuelve, como ya se ha mencionado, la suma absoluta en fragmentos de 8 píxeles. 512 dividido entre 8 es 64, por lo que se pueden sumar en paralelo 8 líneas de 8 bloques de 8x8.

25 La función real que se puede usar es, como se ha mencionado anteriormente, `vp9_avx512_block_average`, que, p. ej., calcula los 64 valores de CC de los 8 bloques de 8x8 en un bloque de 64x64 usando la instrucción SAD de 512 bits. De acuerdo con las mediciones, esta función usa aproximadamente solo el 0,1 % de la carga de CPU del codificador VP9.

30 Volviendo ahora a la figura 3, la figura 3b muestra una imagen de vídeo codificada/decodificada de la figura 3a con información de tipo de bloque. Se decide que los bloques de color gris claro señalados con flechas se codifiquen como 64 x 64 intra, de acuerdo con la presente invención, en contraste con lo que se habría hecho de acuerdo con la codificación/decodificación estándar, p. ej., VP9 sin usar la decisión de acuerdo con la presente invención. En general, la norma VP9 o una similar usa 64 x 64 intra con mucha menos frecuencia en áreas uniformes, dando como resultado más distorsión visual en estas áreas.

35 Las realizaciones del presente documento pueden implementarse en una pluralidad de entidades físicas, tal como una disposición que comprenda dichas una o más entidades físicas, es decir, las realizaciones pueden implementarse de manera distribuida, tal como en un sistema en la nube, que puede comprender un conjunto de máquinas servidoras.

40 Como se usa en el presente documento, los términos "primero", "segundo", "tercero", etc. se han podido usar meramente para distinguir características, aparatos, elementos, unidades o similares entre sí a menos que sea evidente de otro modo a partir del contexto.

45 Como se usa en el presente documento, las expresiones "en algunas realizaciones" y "en determinadas realizaciones" se han usado para indicar que las características de la realización descrita pueden combinarse con cualquier otra realización divulgada en el presente documento.

50 Aunque se han descrito realizaciones de los diversos aspectos, la invención se expone en las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un método de codificación de vídeo implementado en un dispositivo procesador para preparar la determinación de si un bloque de $M \times M$ píxeles en una imagen de vídeo, que se puede dividir en un número entero de $N \times N$ bloques, debería codificarse o no mediante intra-codificación, que comprende las siguientes etapas:
- a) calcular los coeficientes de DCT de los $N \times N$ bloques realizando una primera transformada DCT,
 - b) calcular valores de CC de los coeficientes de DCT de los $N \times N$ bloques
 - c) comparar los coeficientes de DCT de los $N \times N$ bloques con un primer umbral predefinido,
 - d) si todos los coeficientes de DCT de los $N \times N$ bloques están por debajo de un primer umbral predefinido, entonces asignar un "0", si no, asignar un "1", correspondiente a una primera asignación para un primer bloque de $(M/N) \times (M/N)$, caracterizado por
 - e) almacenar todos los valores de coeficiente de CC de los $N \times N$ bloques transformados en un segundo bloque $(M/N) \times (M/N)$, donde cada posición en el segundo bloque $(M/N) \times (M/N)$ contiene el valor CC de un bloque $N \times N$ correspondiente,
 - f) calcular coeficientes de DCT del segundo bloque $(M/N) \times (M/N)$ realizando una segunda transformada DCT,
 - g) si todos los coeficientes de DCT del segundo bloque transformado $(M/N) \times (M/N)$ están por debajo de un segundo umbral predefinido, entonces asignar un "0", si no, asignar un "1", dando como resultado una segunda asignación correspondiente para el segundo bloque $(M/N) \times (M/N)$, y
 - h) si la primera asignación para el primer bloque $(M/N) \times (M/N)$ y la segunda asignación correspondiente para el segundo bloque $(M/N) \times (M/N)$ son cero, entonces se determina que el bloque de $M \times M$ píxeles se va a codificar mediante intra-codificación.
2. El método de acuerdo con la reivindicación 1, caracterizado por que los coeficientes de DCT de los $N \times N$ bloques respectivos se almacenan en una primera máscara de bits de 64 bits.
3. El método de acuerdo con la reivindicación 1, caracterizado por que los respectivos bloques $(M/N) \times (M/N)$ se almacenan en una segunda máscara de bits de 64 bits.
4. El método de acuerdo con la reivindicación 1, caracterizado por que: $M=128$, $M=64$ o $M=32$.
5. El método de acuerdo con la reivindicación 1, caracterizado por que $N=4$, $N=8$, $N=16$ o $N=32$.
6. El método de acuerdo con la reivindicación 1, caracterizado por que las etapas de cálculo se implementan mediante las instrucciones SAD de AVX512 de 512 bits.
7. Un dispositivo procesador de vídeo para codificación de vídeo ajustado para determinar si un bloque de $M \times M$ píxeles en una imagen de vídeo, que se puede dividir en un número entero de $N \times N$ bloques, debería codificarse mediante intra-codificación, en donde el dispositivo procesador de vídeo está ajustado para implementar las siguientes etapas:
- a) calcular los coeficientes de DCT de los $N \times N$ bloques realizando una primera transformada DCT,
 - b) calcular valores de CC de los coeficientes de DCT de los $N \times N$ bloques,
 - c) comparar los coeficientes de DCT de los $N \times N$ bloques con un primer umbral predefinido,
 - d) si todos los coeficientes de DCT de los $N \times N$ bloques están por debajo de un primer umbral predefinido, entonces asignar un "0", si no, asignar un "1", dando como resultado una primera asignación correspondiente para el primer bloque $(M/N) \times (M/N)$,
 - e) almacenar todos los valores de coeficiente de CC de los $N \times N$ bloques transformados en un segundo bloque $(M/N) \times (M/N)$, donde cada posición en el segundo bloque $(M/N) \times (M/N)$ contiene el valor CC de un bloque $N \times N$ correspondiente,
 - f) calcular coeficientes de DCT del segundo bloque $(M/N) \times (M/N)$ realizando una segunda transformada DCT,
 - g) si todos los coeficientes de DCT del segundo bloque transformado $(M/N) \times (M/N)$ están por debajo de un segundo umbral predefinido, entonces asignar un "0", si no, asignar un "1", dando como resultado una segunda asignación correspondiente para el primer bloque $(M/N) \times (M/N)$, y
 - h) si la primera asignación para el primer bloque $(M/N) \times (M/N)$ y la segunda asignación correspondiente para el segundo bloque $(M/N) \times (M/N)$ son cero, entonces se determina que el bloque de $M \times M$ píxeles se va a codificar mediante intra-codificación.
8. El dispositivo procesador de vídeo de acuerdo con la reivindicación 7, caracterizado por que los coeficientes de DCT de los $N \times N$ bloques respectivos se almacenan en una primera máscara de bits de 64 bits.
9. El dispositivo procesador de vídeo de acuerdo con la reivindicación 7, caracterizado por que los respectivos bloques $(M/N) \times (M/N)$ se almacenan en una segunda máscara de bits de 64 bits.

ES 2 964 763 T3

10. El dispositivo procesador de vídeo de acuerdo con la reivindicación 7, caracterizado por que:
M=128, M=64 o M=32.

5 11. El dispositivo procesador de vídeo de acuerdo con la reivindicación 7, caracterizado por que
N=4, N=8, N=16 o N=32.

12. El dispositivo procesador de vídeo de acuerdo con la reivindicación 7, caracterizado por que las etapas de cálculo se implementan por medio de las instrucciones SAD de AVX512 de 512 bits.

10

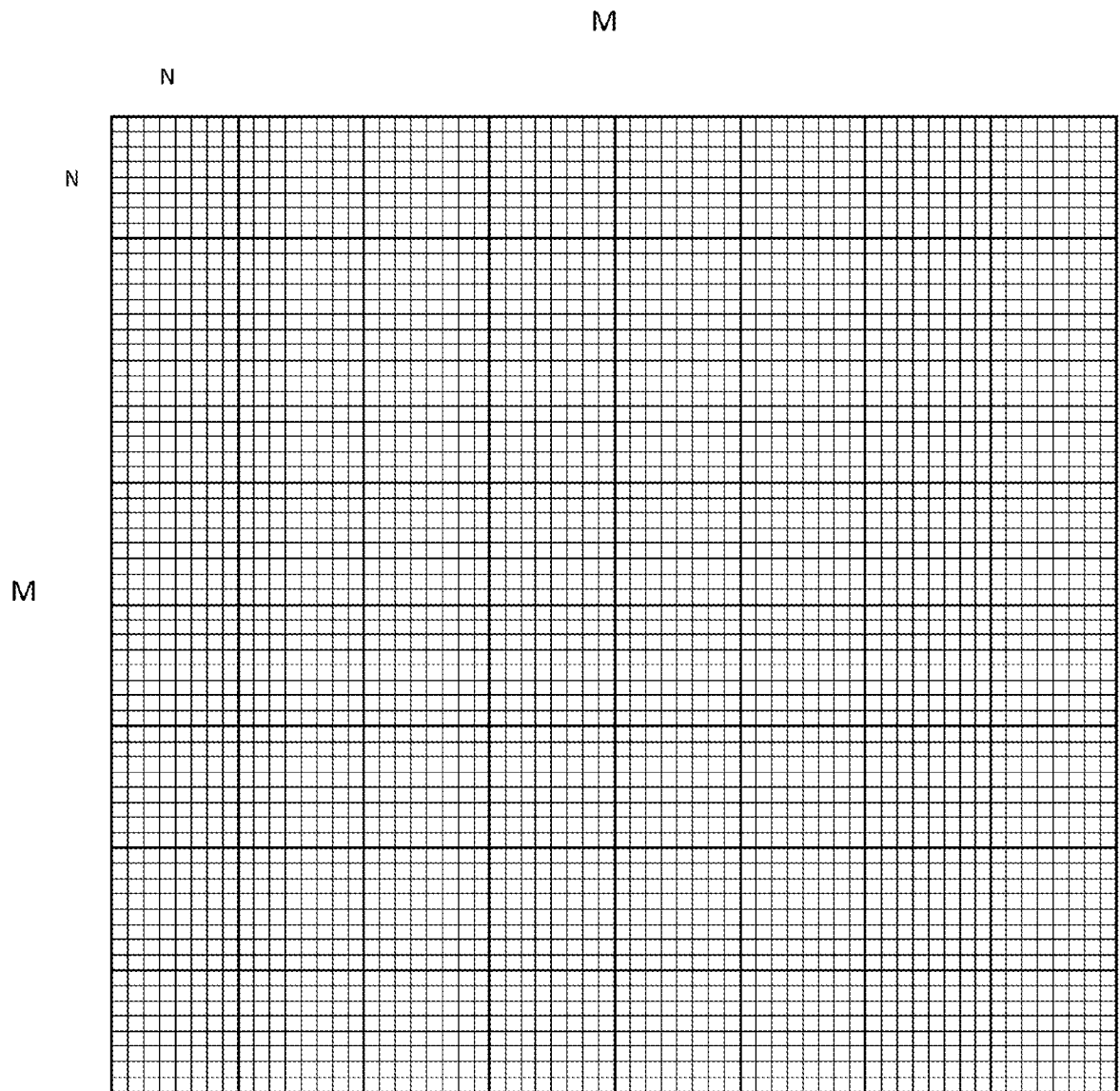


Figura 1

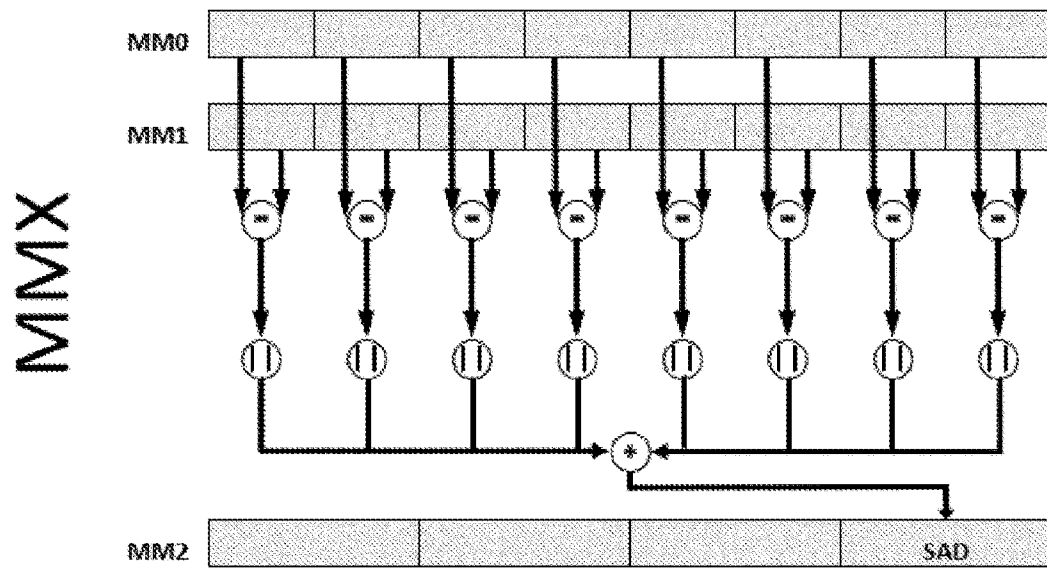


Figura 2a

SSE2

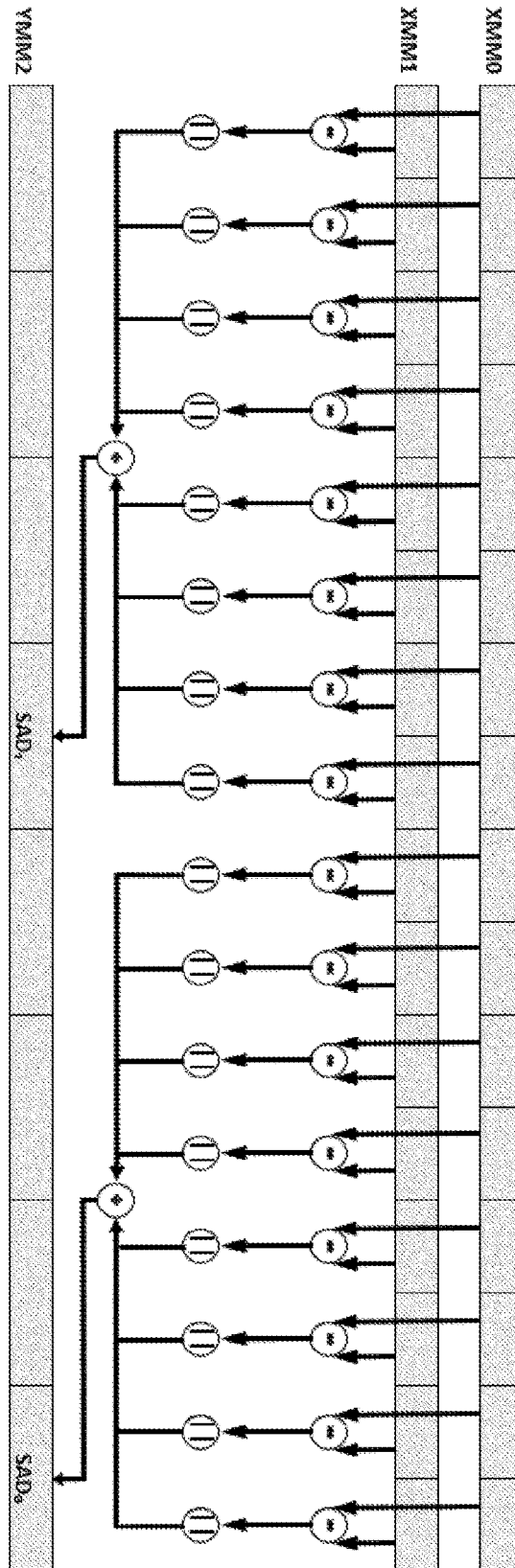


Figura 2b

AVX2

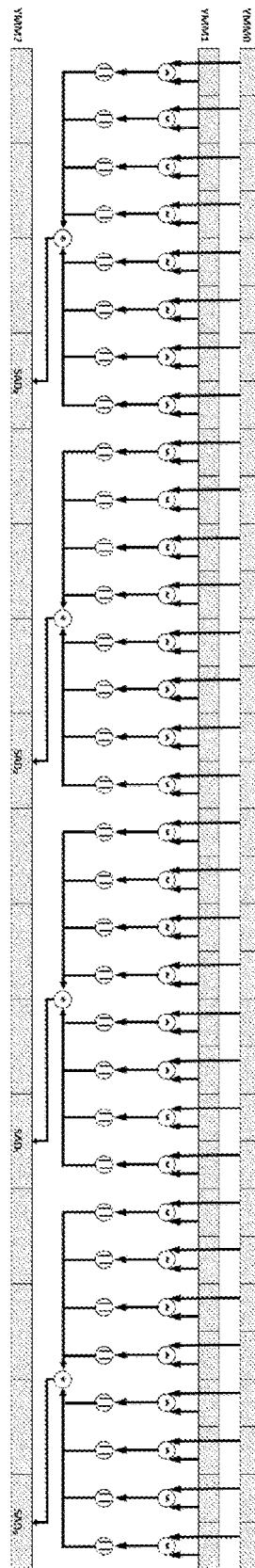


Figura 2c

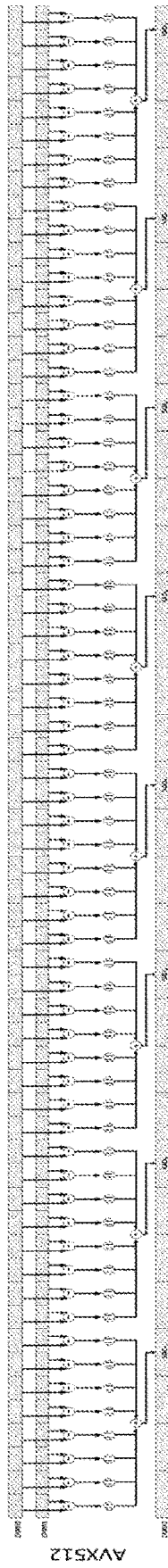


Figura 2d



Figura 3a

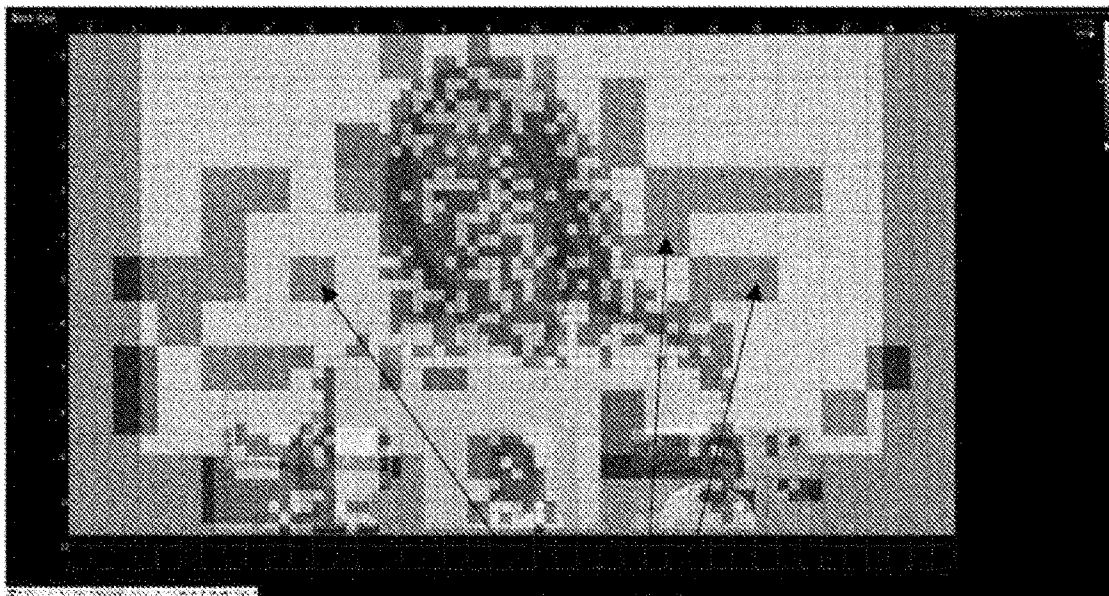


Figura 3b