

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 776 196**

51 Int. Cl.:

G06K 9/46 (2006.01)

G06T 7/40 (2007.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **16.07.2014 PCT/IB2014/063158**

87 Fecha y número de publicación internacional: **29.01.2015 WO15011612**

96 Fecha de presentación y número de la solicitud europea: **16.07.2014 E 14777780 (9)**

97 Fecha y número de publicación de la concesión europea: **08.01.2020 EP 3025274**

54 Título: **Aparato de procesamiento de imágenes y método para codificar un descriptor de imágenes basándose en un histograma de gradiente**

30 Prioridad:

24.07.2013 IT TO20130629

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

29.07.2020

73 Titular/es:

**NEW LUCK GLOBAL LIMITED (100.0%)
P.O. Box 957, Offshore Incorporations Centre
Road Town, Tortoia, VG**

72 Inventor/es:

PASCHALAKIS, STAVROS

74 Agente/Representante:

FÚSTER OLAGUIBEL, Gustavo Nicolás

ES 2 776 196 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

Aparato de procesamiento de imágenes y método para codificar un descriptor de imágenes basándose en un histograma de gradiente

5

Descripción

La presente invención se refiere a un método para codificar, transcodificar, decodificar y procesar eficazmente descriptores de imágenes calculados en regiones locales en torno a puntos clave de interés de imágenes y a un dispositivo de procesamiento de imágenes que comprende medios para codificar, transcodificar, decodificar y procesar tales descriptores.

10

Tales descriptores de imágenes han encontrado una amplia aplicabilidad en muchas aplicaciones de visión por ordenador, incluyendo reconocimiento de objetos, recuperación de imágenes basándose en contenido y registro de imágenes, por nombrar algunos.

15

Los enfoques existentes para la codificación de tales descriptores presentan determinados inconvenientes.

Por ejemplo, los enfoques de codificación existentes dan como resultado descriptores que requieren el análisis de todos los descriptores para realizar transcodificación, por lo que un descriptor de una longitud de descriptor dada se convierte en un descriptor de una longitud de descriptor diferente, o para realizar decodificación y comparación de descriptores de diferentes longitudes.

20

Un ejemplo adicional se facilita en Mehmet Firat VURAL *et al.* en "Registration of Multispectral Satellite Images with Orientation-Restricted SIFT" publicado en el simposio internacional de geociencias y teledetección 2009 del IEEE, IGARSS 2009, IEEE, PISCATAWAY, NJ, EE.UU., (2009-07-2012), donde se describe el algoritmo SIFT de orientación restringida (OR-SIFT). OR-SIFT es una versión modificada del popular algoritmo SIFT; sin embargo, sólo reduce a la mitad la longitud de los descriptores SIFT, por lo que no logra una buena escalabilidad de codificación de un descriptor de imágenes basándose en el algoritmo SIFT.

25

30

Como otro ejemplo, los enfoques de codificación existentes son ineficaces en lo que se refiere a la complejidad de codificación porque ignoran las coincidencias y redundancias en las operaciones que se requieren para producir descriptores de imágenes de longitud variable.

35

La solicitud de patente italiana n.º T02012A000602 (publicada como documento WO 2014/009198 A1) presentada por el propio solicitante describe la codificación de descriptores de imágenes locales, mediante lo cual descriptores de imágenes robustos, discriminativos, escalables y compactos se calculan a partir de descriptores de imágenes que emplean histogramas de gradientes basándose en la transformación de dichos histogramas de gradientes, donde dicha transformación captura información relevante y robusta contenida en ellos en cuanto a la forma de las distribuciones y la relación entre sus valores de compartimento.

40

En dicha solicitud de patente italiana se divulgan métodos de codificación de dichos descriptores que son más eficaces que los métodos de la técnica anterior en lo que se refiere a producir flujos de bits fácilmente escalables.

45

Tales descriptores se divulgan en la solicitud de patente italiana aún no publicada mencionada anteriormente n.º TO2012A000602 que divulga el cálculo de descriptores de imágenes robustos, discriminativos, escalables y compactos a partir de descriptores de imágenes que emplean histogramas de gradientes basándose en la transformación de dichos histogramas de gradientes, donde dicha transformación captura la información relevante y robusta contenida en ellos en cuanto a la forma de las distribuciones y la relación entre sus valores de compartimento.

50

A continuación en el presente documento se describen aspectos importantes del cálculo de descriptores de imágenes robustos, discriminativos, escalables y compactos a partir de descriptores de imágenes que emplean histogramas de gradientes, en particular un descriptor de imágenes SIFT, según la solicitud de patente italiana aún no publicada n.º TO2012A000602.

55

Brevemente, con el método SIFT, se forman descriptores de imágenes locales de la siguiente manera: en primer lugar, se realiza una búsqueda en múltiples escalas y ubicaciones de imágenes para identificar y localizar puntos clave de imágenes estables que sin invariantes en escala y orientación; entonces, para cada punto clave, se determinan una o más orientaciones dominantes basándose en gradientes de imágenes locales, lo que permite que el cálculo del descriptor local posterior se realice en relación con la orientación, escala y ubicación asignadas de cada punto clave, logrando de ese modo la invariancia para estas transformaciones. Entonces, se forman descriptores de imágenes locales en torno a puntos clave de la siguiente manera: en primer lugar, se calculan la magnitud del gradiente y la información de orientación en los puntos de muestra de imágenes en una región en torno al punto clave; entonces, estas muestras se acumulan en histogramas de orientación que resumen el contenido a lo largo de $n \times n$ subregiones.

60

65

5 A modo de ilustración únicamente, se muestra un ejemplo de un descriptor de puntos clave SIFT en las figuras 1a y 1b, donde la figura 1a muestra una subdivisión de una región local R en 4x4 subregiones SR y la figura 1b muestra una subdivisión del rango de 360° de orientaciones en ocho compartimentos para cada histograma de orientación, correspondiendo la longitud de cada flecha a la magnitud de esa entrada de histograma. Por tanto, un descriptor de imágenes local tal como se ilustra en la figura 1 tiene $4 \times 4 \times 8 = 128$ elementos. Pueden encontrarse más detalles de la técnica SIFT en David G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, 60, 2 (2004), págs. 91-110.

10 Según la solicitud de patente italiana aún no publicada n.º TO2012A000602, puede calcularse un descriptor de imágenes robusto, discriminativo, escalable y compacto a partir de un descriptor SIFT de la siguiente manera.

15 En la siguiente descripción, H es un descriptor SIFT completo que comprende 16 histogramas de gradientes h cada uno con ocho compartimentos h, mientras que V es un descriptor local completo según la presente invención que comprende 16 subdescriptores v cada uno con ocho elementos v.

20 Supóngase que H indica un descriptor de imágenes local SIFT que comprende 16 histogramas de gradientes $h_0 - h_{15}$, tal como se muestra en la figura 2a, comprendiendo cada histograma ocho valores de compartimento $h_0 - h_7$, tal como se muestra en la figura 2b. Puede calcularse un descriptor de imágenes más robusto, discriminativo, escalable y compacto transformando cada uno de $h_0 - h_{15}$ de H y luego realizando cuantificación escalar en los valores transformados resultantes. Más específicamente, cada uno de $h_0 - h_{15}$ se transforma según la transformada A o la transformada B, tal como se muestra a continuación, según la información de utilización de transformadas de la figura 3, es decir, la transformada A se aplica a $h_0, h_2, h_5, h_7, h_8, h_{10}, h_{13}, h_{15}$ y la transformada B se aplica a $h_1, h_3, h_4, h_6, h_9, h_{11}, h_{12}, h_{14}$, dando el descriptor transformado V con subdescriptores $v_0 - v_{15}$, correspondientes a $h_0 - h_{15}$ respectivamente, y comprendiendo cada uno elementos $v_0 - v_7$, dando un total de 128 elementos.

Transformada A (1)

$$\begin{aligned} v_0 &= h_2 - h_6 \\ v_1 &= h_3 - h_7 \\ v_2 &= h_0 - h_1 \\ v_3 &= h_2 - h_3 \\ v_4 &= h_4 - h_5 \\ v_5 &= h_6 - h_7 \\ v_6 &= (h_0 + h_4) - (h_2 + h_6) \\ v_7 &= (h_0 + h_2 + h_4 + h_6) - (h_1 + h_3 + h_5 + h_7) \end{aligned}$$

Transformada B (2)

$$\begin{aligned} v_0 &= h_0 - h_4 \\ v_1 &= h_1 - h_5 \\ v_2 &= h_7 - h_0 \\ v_3 &= h_1 - h_2 \\ v_4 &= h_3 - h_4 \\ v_5 &= h_5 - h_6 \\ v_6 &= (h_1 + h_5) - (h_3 + h_7) \\ v_7 &= (h_0 + h_1 + h_2 + h_3) - (h_4 + h_5 + h_6 + h_7) \end{aligned}$$

30 Entonces, cada elemento se somete a cuantificación escalar aproximada, por ejemplo, cuantificación ternaria (3 niveles), con los umbrales de cuantificación seleccionados para lograr una distribución de probabilidad de aparición específica entre los compartimentos de cuantificación para cada elemento. Esta cuantificación escalar produce el descriptor cuantificado V_{\sim} con los subdescriptores $v_{\sim 0} - v_{\sim 15}$, comprendiendo cada uno los elementos $v_{\sim 0} - v_{\sim 7}$, de nuevo con un total de 128 elementos. Este descriptor compacto captura la información más discriminativa y robusta contenida en los histogramas de gradientes originales, en cuanto a la forma de las distribuciones y la relación entre sus valores de compartimento.

40 Una ventaja clave del descriptor V_{\sim} , así como de su versión cuantificada V_{\sim} , es que es altamente escalable, y su dimensionalidad puede reducirse fácilmente si lo requieren los requisitos de almacenamiento de una aplicación o las características de un canal de transmisión eliminando simplemente uno o más de sus elementos. Por motivos de

simplicidad, en la descripción que sigue se describirán aspectos importantes de la invención en lo que se refiere a codificación del descriptor previamente cuantificado V con subdescriptores $v_0 - v_{15}$, comprendiendo cada uno elementos v_0-v_7 y, a menos que se indique lo contrario, debe entenderse que la codificación del descriptor cuantificado V , se desarrolla de manera similar.

5 Las figuras 4a-4e muestran conjuntos de elementos a modo de ejemplo que se ha encontrado que producen excelente poder discriminativo y robustez para cinco longitudes de descriptor objetivo, desde la longitud de descriptor 0 (DL0), la longitud de descriptor más corta utilizando sólo 20 elementos de descriptor, hasta la longitud de descriptor 4 (DL4), la longitud de descriptor más larga utilizando los 128 elementos. Más específicamente, la figura 4a muestra un conjunto de elementos a modo de ejemplo para la longitud de descriptor DL0 que comprende 20 elementos, la figura 4b muestra un conjunto de elementos a modo de ejemplo para la longitud de descriptor DL1 que comprende 40 elementos, la figura 4c muestra un conjunto de elementos a modo de ejemplo para la longitud de descriptor DL2 que comprende 64 elementos, la figura 4d muestra un conjunto de elementos a modo de ejemplo para la longitud de descriptor DL3 que comprende 80 elementos, y la figura 4e muestra un conjunto de elementos a modo de ejemplo para la longitud de descriptor DL4 que comprende los 128 elementos. Por tanto, para cada longitud de descriptor, cada elemento de cada subdescriptor se codificará o no según los conjuntos de utilización de elementos de la figura 4a-4e.

20 Es clave para esta propiedad de escalabilidad que el conjunto de elementos utilizados para cada longitud de descriptor debe ser igual que o un subconjunto del conjunto de elementos utilizados para todas las longitudes de descriptor más altas, tal como se ilustra en las figuras 4a-4e. Esto permite la transcodificación y la comparación de descriptores de diferentes longitudes mediante la simple eliminación de elementos en exceso del descriptor con la longitud de descriptor más alta, de modo que se reduce al mismo conjunto de elementos que el descriptor con la longitud de descriptor más baja.

25 Un método de codificación directo de este descriptor comprende calcular y codificar los elementos en un orden "por subdescriptor", es decir, en el caso general como $v_{0,0}, v_{0,1}, \dots, v_{0,7}, v_{1,0}, v_{1,1}, \dots, v_{1,7}, \dots, v_{15,0}, v_{15,1}, \dots, v_{15,7}$ donde v_{ij} indica el elemento v_j del subdescriptor v_i . Esto significa elementos de codificación v_0, v_1, \dots, v_7 para el subdescriptor transformado v_0 , luego elementos de codificación v_0, v_1, \dots, v_7 para el subdescriptor transformado v_1 , etc., usando las transformadas apropiadas, por ejemplo, tal como se ilustra en la figura 3, y también usando los conjuntos de utilización de elementos apropiados para la longitud de descriptor deseada, por ejemplo, tal como se ilustra en la figura 4, para decidir qué elementos deben codificarse.

35 Esta codificación resulta, por ejemplo para una longitud de descriptor DL0, en un descriptor $v_{0,0}, v_{1,0}, v_{2,0}, v_{3,0}, v_{4,0}, v_{5,0}, v_{5,6}, v_{6,0}, v_{6,6}, v_{7,0}, v_{8,0}, v_{9,0}, v_{9,6}, v_{10,0}, v_{10,6}, v_{11,0}, v_{12,0}, v_{13,0}, v_{14,0}, v_{15,0}$ y para una longitud de descriptor DL1 en un descriptor $v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1}, v_{2,0}, v_{2,1}, v_{3,0}, v_{3,1}, v_{4,0}, v_{4,1}, v_{5,0}, v_{5,1}, v_{5,2}, v_{5,6}, v_{6,0}, v_{6,1}, v_{6,2}, v_{6,6}, v_{7,0}, v_{7,1}, v_{8,0}, v_{8,1}, v_{9,0}, v_{9,1}, v_{9,2}, v_{9,6}, v_{10,0}, v_{10,1}, v_{10,2}, v_{10,6}, v_{11,0}, v_{11,1}, v_{12,0}, v_{12,1}, v_{13,0}, v_{13,1}, v_{14,0}, v_{14,1}, v_{15,0}, v_{15,1}$.

40 La figura 5 ilustra el funcionamiento de un codificador tan directo como una secuencia de etapas. En la siguiente descripción, así como en descripciones posteriores del funcionamiento de un codificador, a menos que se especifique lo contrario, una secuencia de etapas de este tipo corresponde a etapas que son conceptuales y no corresponden a hardware específico de implementaciones de software, componentes e instrucciones, pero son representativas del funcionamiento general del codificador. Más específicamente, la figura 5 ilustra el funcionamiento de un codificador para una longitud de descriptor DLk, por ejemplo correspondiente a una de las longitudes de descriptor ilustradas en la figura 4. En la etapa S100 de la figura 5, la codificación del descriptor comienza en el primer subdescriptor, es decir v_0 . En la etapa S110, se selecciona la transformada apropiada para el subdescriptor que está procesándose, por ejemplo, según la utilización de transformadas de la figura 3. Cabe señalar que el cálculo del descriptor V a partir del descriptor H según dos transformadas diferentes tal como se describe en este caso es sólo un ejemplo. El cálculo del descriptor V a partir del descriptor H también puede realizarse según una sola transformada, por ejemplo, sólo la transformada A o sólo la transformada B, haciendo innecesaria la etapa S110, o según más de dos transformadas. En la etapa S120, la codificación del subdescriptor que está procesándose comienza en el primer elemento del subdescriptor, es decir, v_0 . Entonces, en la etapa S130, se comprueba el uso o no del elemento particular del subdescriptor particular, es decir $v_{0,0}$, frente a la información de utilización del elemento para la longitud de descriptor DLk, por ejemplo, usando uno de los conjuntos de utilización de la figura 4. Si el elemento no está en uso, el procesamiento avanza entonces a la etapa S150. Si el elemento está en uso para la longitud de descriptor DLk, entonces tiene lugar su codificación en la etapa S140. En este caso, así como en las descripciones posteriores del funcionamiento de un codificador, a menos que se especifique lo contrario, la palabra "codificación" significa una o más acciones, o combinación de las mismas, que hacen del elemento $v_{0,0}$ parte del descriptor de imágenes local, incluyendo dichas acciones, a modo de ejemplo y sin limitación, el cálculo según la función de transformada apropiada de (1) o (2) indicadas anteriormente, la selección del elemento para su inclusión en el descriptor de imágenes local en el caso de que todos los elementos se calculen previamente sin saber qué elementos se usarán finalmente en el descriptor, la cuantificación del valor del elemento, el almacenamiento del elemento en la memoria volátil o no volátil y la transmisión del elemento a lo largo de un canal de transmisión. Después de la etapa S140, o si se decidió que el elemento no está en uso para la longitud de descriptor DLk en la etapa S130, el procesamiento avanza a la etapa S150. En la etapa S150, si el elemento actual no es el último elemento del subdescriptor, el procesamiento avanza al siguiente elemento del subdescriptor; en

caso contrario, el procesamiento avanza a la etapa S160. En la etapa S160, si el subdescriptor actual no es el último subdescriptor del descriptor de imágenes local, el procesamiento avanza al siguiente subdescriptor del descriptor de imágenes local, en caso contrario el procesamiento finaliza. Por tanto, está claro que las etapas S100, S120, S150 y S160 se refieren al orden en que se realiza el procesamiento, mientras que las etapas S110, S130 y S140 se refieren a la codificación real del descriptor de imágenes local.

Otro método de codificación directo de este descriptor comprende calcular y codificar los elementos en un orden "por elemento", es decir, en el caso general como $v_{0,0}, v_{1,0}, \dots, v_{15,0}, v_{0,1}, v_{1,1}, \dots, v_{15,1}, \dots, v_{0,7}, v_{1,7}, \dots, v_{15,7}$ es decir, elemento de codificación v_0 para los subdescriptores v_0, v_1, \dots, v_{15} , luego el elemento de codificación v_1 para los subdescriptores v_0, v_1, \dots, v_{15} , etc., nuevamente usando las transformadas apropiadas, por ejemplo, tal como se ilustra en la figura 3, y también usando los conjuntos de utilización de elementos apropiados para la longitud de descriptor deseada, por ejemplo, tal como se ilustra en la figura 4, para decidir qué elementos deben codificarse. Un codificador de este tipo puede funcionar de manera análoga al codificador de la figura 5, con la reordenación apropiada de las etapas. En general, ninguno de los dos métodos mencionados anteriormente ofrece una ventaja con respecto al otro método. Para los fines de transcodificación, decodificación y procesamiento, el decodificador también debe conocer el proceso de codificación y los conjuntos de ordenación y utilización de elementos para poder procesar y comparar descriptores, posiblemente de diferentes longitudes, para los fines de las aplicaciones de visión por ordenador relacionadas. Por tanto, los conjuntos de utilización de elementos o bien deben fijarse de manera permanente o bien almacenarse/transmitirse junto con los descriptores. En este contexto, el proceso de codificación directo es desventajoso.

Más específicamente, una codificación de este tipo ignora la importancia relativa entre diferentes elementos en el orden de codificación. En consecuencia, en lo que se refiere a la transcodificación, mediante la cual un descriptor de una longitud de descriptor dada se convierte en un descriptor de una longitud de descriptor diferente, o en lo que se refiere a la decodificación y comparación de descriptores de diferentes longitudes comparando elementos correspondientes entre los dos descriptores, tal codificación necesita análisis de los descriptores para lograr el resultado deseado.

Además, tal codificación ignora los patrones de redundancia en la importancia relativa entre diferentes elementos y es innecesariamente compleja en lo que respecta a decidir si deben codificarse o no elementos específicos.

Por tanto un objeto de la presente invención es divulgar un método para codificar un descriptor de imágenes basándose en un histograma de gradientes, y un aparato de procesamiento de imágenes relativo, que sea más eficaz con respecto a los métodos de la técnica anterior.

Un objeto adicional de la presente invención es divulgar un método para codificar un descriptor de imágenes basándose en un histograma de gradientes, y un aparato de procesamiento de imágenes relativo, que sea más flexible.

Un objeto adicional de la presente invención es divulgar un método para codificar un descriptor de imágenes basándose en un histograma de gradientes, y un aparato de procesamiento de imágenes relativo, que permita lograr una implementación de codificador optimizada.

Un objeto adicional de la presente invención es divulgar un método para codificar un descriptor de imágenes basándose en un histograma de gradientes, y un aparato de procesamiento de imágenes relativo, que permita obtener descriptores de imágenes de cualquier longitud.

Estos y otros objetos de la invención se logran a través de un método para codificar un descriptor de imágenes basándose en un histograma de gradientes, y un aparato de procesamiento de imágenes relativo, tal como se reivindica en las reivindicaciones adjuntas, que constituyen una parte integral de la presente descripción.

Brevemente, se divulga un método para codificar eficazmente descriptores de imágenes tales como los descritos anteriormente codificándolos según un orden de utilización de elementos, lo que da como resultado descriptores escalables que pueden convertirse en longitudes de descriptor más bajas mediante el simple truncamiento del descriptor en lugar de análisis.

La codificación se lleva a cabo según grupos de subdescriptores, formados según los patrones de redundancia en la importancia relativa entre los elementos correspondientes de dichos grupos de subdescriptores.

Más específicamente, dicha agrupación se realiza agrupando subdescriptores cuyos elementos correspondientes tienen una importancia similar en una ordenación de todos los elementos del descriptor según su importancia relativa en lo que se refiere a lograr un alto rendimiento de reconocimiento, más específicamente agrupando subdescriptores según su distancia desde el centro del descriptor y, adicionalmente, agrupando subdescriptores según la distancia entre ellos y/u ordenando los subdescriptores de un grupo según las características de codificación correspondientes y/u adicionalmente ordenando los subdescriptores de un grupo según la distancia entre ellos.

El método de codificación según la invención es ventajosamente más eficaz que los de la técnica anterior en lo que se refiere a eficacia, complejidad computacional y/o la cantidad de información necesaria para generar flujos de bits escalables.

5 Características adicionales de la invención se exponen en las reivindicaciones adjuntas, que pretenden ser una parte integral de la presente descripción.

10 Los objetos anteriores resultarán más evidentes a partir de la siguiente descripción detallada del método para codificar un descriptor de imágenes basándose en un histograma de gradientes, y un aparato de procesamiento de imágenes relativo, con referencia particular a los dibujos adjuntos, en los que:

- las figuras 1a y 1b muestran un ejemplo de un descriptor de puntos clave de la técnica anterior;
- 15 - las figuras 2a y 2b muestran histogramas de gradientes del descriptor de puntos clave de la figura 1 y valores de compartimento relacionados con uno de dichos histogramas de gradientes, respectivamente;
- la figura 3 muestra transformadas a modo de ejemplo que van a aplicarse a los histogramas de gradientes de la figura 2;
- 20 - las figuras 4a-4e muestran conjuntos de elementos a modo de ejemplo para cinco longitudes de descriptor objetivo respectivas;
- la figura 5 representa un diagrama de flujo que ilustra el funcionamiento de un codificador que usa los conjuntos de elementos de la figura 4;
- 25 - la figura 6 muestra un orden de utilización de elementos usado por un método según la presente invención;
- la figura 7 ilustra el funcionamiento de un codificador que usa el orden de utilización de elementos de la figura 6;
- 30 - la figura 8 muestra los centros de una región y de una subregión de un descriptor de imágenes local;
- la figura 9 muestra una primera agrupación de subdescriptores de un descriptor de imágenes local según una primera realización o una cuarta realización de un método según la invención;
- 35 - las figuras 12 y 14 muestran las agrupaciones a modo de ejemplo segunda y tercera, respectivamente, de subdescriptores de un descriptor de imágenes local según la primera realización de un método según la invención;
- las figuras 10, 13 y 15 muestran órdenes de utilización de elementos de los grupos primero, segundo y tercero en relación con las agrupaciones de la figuras 9, 12, 14 y 17, respectivamente;
- 40 - la figura 11 ilustra el funcionamiento de un codificador según una primera, una segunda o una cuarta realización de un método según la invención;
- 45 - la figura 16 muestra una cuarta agrupación a modo de ejemplo según una primera realización de un método según la invención;
- la figura 17 muestra una quinta agrupación a modo de ejemplo de subdescriptores de un descriptor de imágenes local según la segunda y la tercera realización de un método según la invención;
- 50 - la figura 18 ilustra el funcionamiento de un codificador que usa el orden de utilización de elementos de la figura 17, según la tercera realización de un método según la invención;
- la figura 19 ilustra el funcionamiento de un codificador según una cuarta realización de un método según la invención;
- 55 - la figura 20 muestra una agrupación a modo de ejemplo de subdescriptores de un descriptor de imágenes local según una quinta realización de un método según la invención;
- 60 - las figuras 21a-21e muestran conjuntos de elementos a modo de ejemplo según la agrupación de la figura 20;
- las figuras 22a-22e muestran los conjuntos de elementos de la figuras 21a-21e, respectivamente, cuando se convierten en conjuntos de utilización de elementos de grupo;
- 65 - la figura 23 ilustra el funcionamiento de un codificador para la codificación de un descriptor para proporcionar los conjuntos de elementos convertidos de la figuras 22a-22e;

- la figura 24 ilustra un dispositivo de procesamiento de imágenes adecuado para llevar a cabo el método según la presente invención.

5 Según la invención, en lugar de codificar según los conjuntos de utilización de elementos para diferentes longitudes de descriptor, un codificador más eficaz puede funcionar según un orden de utilización de elementos, produciendo un descriptor cuyos elementos se ordenan según el orden de utilización de elementos y que pueden convertirse en longitudes de descriptor más bajas mediante el simple truncamiento del descriptor. Tal orden de utilización de elementos puede adoptar la forma de una lista ordenada de 128 elementos, que puede codificarse en 112 bytes, especificando cada entrada de lista un índice de subdescriptor y un índice de elemento, tal como se ilustra en la figura 6. Por tanto, por ejemplo, la figura 6 muestra una lista de prioridad de elementos mediante la cual al elemento $v_{5,0}$ se le da la prioridad más alta, al elemento $v_{9,0}$ se le da la segunda prioridad más alta, etc. Por tanto, un codificador de orden de utilización de elementos de este tipo puede producir un descriptor de longitud l codificando los l elementos superiores en la lista.

15 La figura 7 ilustra el funcionamiento de un codificador de este tipo que usa el orden de utilización de elementos de la figura 6. Con el codificador de la figura 7, la codificación del descriptor comienza con el elemento con la máxima prioridad (prioridad 1) en el orden de utilización de elementos, es decir, el elemento v_0 del subdescriptor v_5 . En la etapa S200, se selecciona la transformada apropiada según el subdescriptor al que pertenece este elemento, por ejemplo, según la utilización de transformadas de la figura 3. Cabe señalar que el cálculo del descriptor V a partir del descriptor H según dos transformadas diferentes tal como se describe en este caso es sólo un ejemplo. En diferentes realizaciones, el cálculo del descriptor V a partir del descriptor H también puede realizarse según una sola transformada, por ejemplo, sólo la transformada A o sólo la transformada B, haciendo innecesaria la etapa S200, o según más de dos transformadas. Entonces, tiene lugar la codificación del elemento, es decir $v_{5,0}$, en la etapa S210. Luego, en la etapa S220, si un número deseado de l elementos del descriptor aún no se han codificado, el procesamiento avanza al elemento con la siguiente prioridad más alta en el orden de utilización de elementos; en caso contrario, el procesamiento finaliza. Por tanto, la etapa S220 se refiere al control de cuántos elementos codificar, mientras que las etapas S200 y S210 se refieren a la codificación real del descriptor de imágenes local.

20 Por tanto, el codificador de la figura 7, usando un orden de utilización de elementos tal como en la figura 6, produce descriptores cuyos elementos se ordenan según el orden de utilización de elementos y que pueden convertirse en longitudes de descriptor más bajas mediante el simple truncamiento del descriptor, es decir, eliminando los últimos elementos del descriptor, y es más flexible que el codificador de la figura 5, que usa conjuntos de utilización de elementos como en la figura 4.

35 La cantidad de elementos l en el descriptor puede almacenarse/transmitirse junto con el descriptor, posiblemente a nivel de imagen. Para los fines de transcodificación, decodificación y procesamiento, el decodificador también debe conocer el orden de los elementos para poder procesar para los fines de las aplicaciones de visión por ordenador relacionadas. Por tanto, el orden de utilización de elementos o bien debe fijarse de manera permanente o bien almacenarse/transmitirse junto con los descriptores.

40 En la práctica, sin embargo, la eficacia del codificador de la figura 7 puede mejorarse. Esto se debe a que el orden de utilización de elementos ilustrado en la figura 6 contiene una cantidad significativa de redundancia y, en cierta medida, no resulta práctico de generar y usar.

45 La razón de esto es que el conjunto completo de 128 elementos de V no corresponde a un solo descriptor cuyos elementos pueden ordenarse según una lista de prioridad única, sino a 16 subdescriptores de 8 elementos diferentes, extraído cada subdescriptor de un histograma de gradientes diferente según una transformada específica y de modo que los elementos correspondientes entre todos los subdescriptores capturen la relación entre compartimentos con la misma separación angular.

50 Por tanto, se ha encontrado que, en lo que se refiere a lograr un alto rendimiento de reconocimiento con un conjunto de elementos limitado, un descriptor necesita encontrar un equilibrio entre una distribución de elementos uniforme, es decir, seleccionando el/los elemento(s) de tantos subdescriptores como sea posible, y la distancia desde el centro del descriptor, es decir, dando mayor prioridad a los subdescriptores que están más cerca del centro del descriptor. Al mismo tiempo, también se ha encontrado que la importancia de elementos correspondientes de diferentes subdescriptores es aproximadamente la misma cuando la distancia de los subdescriptores al centro del descriptor es la misma, mientras que la importancia de elementos correspondientes de diferentes subdescriptores aumenta cuando disminuye la distancia de los subdescriptores al centro del descriptor.

60 En este contexto, la distancia de un subdescriptor v al centro del descriptor se refiere a la distancia entre el centro de la subregión que corresponde al histograma de gradientes h que da lugar al subdescriptor v y el centro de la región que comprende las subregiones, tal como se ilustra en la figura 8. Aunque es posible referirse nuevamente a las dimensiones de la región y las subregiones de la imagen para calcular dichas distancias, no es necesario ya que sólo interesa comparar dichas distancias. Por tanto, dichas distancias pueden calcularse suponiendo que cada lado de cada subregión tiene, por ejemplo, una longitud unitaria. Además, en esta descripción, dichas distancias son

distancias euclidianas, aunque también pueden usarse otras medidas de distancia adecuadas.

En consecuencia, se observa que, para los fines de la codificación eficaz de un descriptor de imágenes, sus subdescriptores pueden agruparse de modo que a elementos de descriptor correspondientes para los subdescriptores de cada grupo se les asigne una importancia común en el orden de utilización y se codifiquen conjuntamente.

Primera realización

En una primera realización de la invención, los subdescriptores se agrupan según su distancia desde el centro del descriptor.

Por ejemplo, uno de tales agrupamientos se ilustra en la figura 9, donde hay tres grupos, concretamente $g_0 = \{v_0, v_3, v_{12}, v_{15}\}$, que contiene los subdescriptores con la mayor distancia al centro del descriptor, $g_1 = \{v_1, v_2, v_4, v_7, v_8, v_{11}, v_{13}, v_{14}\}$, que contiene los subdescriptores con la segunda mayor distancia al centro del descriptor, y $g_2 = \{v_5, v_6, v_9, v_{10}\}$, que contiene los subdescriptores con la menor distancia al centro del descriptor. Dentro de cada grupo, los subdescriptores se ordenan en orden de índice de subdescriptor ascendente, aunque esto no es restrictivo y pueden usarse otros órdenes, tal como en sentido horario, comenzando por el subdescriptor más a la izquierda superior del grupo. Dentro de cada grupo, a todos los elementos correspondientes de los subdescriptores del grupo se les asigna la misma prioridad de codificación.

Basándose en esta agrupación, puede generarse un orden de utilización de elementos de grupo, que puede adoptar la forma de una lista ordenada de 24 elementos, que puede codificarse en 15 bytes, especificando cada entrada de lista un grupo de subdescriptores y un índice de elemento, tal como se ilustra en la figura 10. Cabe señalar que el orden de utilización de elementos de grupo de la figura 10 es simplemente un ejemplo, y pueden generarse diferentes órdenes de utilización de elementos de grupo cambiando las prioridades de las entradas en la lista. Por tanto, el orden de utilización de elementos de grupo de la figura 10 es mucho más económico que el orden de utilización de elementos de la figura 6, en lo que se refiere a la longitud y el tamaño de codificación. Por tanto, por ejemplo, la figura 10 muestra una lista de prioridades de elementos de grupo mediante la cual al elemento v_0 del grupo g_2 se le da la prioridad más alta, indicando al codificador que los primeros cuatro elementos para codificar son $v_{5,0}, v_{6,0}, v_{9,0}$ y $v_{10,0}$, al elemento v_0 del grupo g_1 se le da la segunda prioridad más alta, indicando al codificador que los siguientes ocho elementos para codificar son $v_{1,0}, v_{2,0}, v_{4,0}, v_{7,0}, v_{8,0}, v_{11,0}, v_{13,0}$ y $v_{14,0}$, etc.

La figura 11 ilustra el funcionamiento de un codificador de este tipo que usa el orden de utilización de elementos de grupo de la figura 10 y configurado para codificar los m grupos superiores de elementos en dicho orden de utilización de elementos de grupo. Con el codificador de la figura 11, la codificación del descriptor comienza con el grupo de elementos con la máxima prioridad (prioridad 1) en el orden de utilización de elementos de grupo, es decir, el elemento v_0 del grupo g_2 que comprende los subdescriptores v_5, v_6, v_9 y v_{10} . En la etapa S300, la codificación del descriptor comienza en el primer subdescriptor del grupo, es decir, v_5 . En la etapa S310, se selecciona la transformada apropiada para el subdescriptor, por ejemplo, según la utilización de transformadas de la figura 3. Cabe señalar que el cálculo del descriptor V a partir del descriptor H según dos transformadas diferentes tal como se describe en este caso es sólo un ejemplo. En diferentes realizaciones, el cálculo del descriptor V a partir del descriptor H también puede realizarse según una sola transformada, por ejemplo sólo la transformada A o sólo la transformada B , haciendo innecesaria la etapa S310, o según más de dos transformadas. Entonces, tiene lugar la codificación del elemento, es decir $v_{5,0}$, en la etapa S320. En la etapa S330, si el subdescriptor actual no es el último subdescriptor en el grupo, el procesamiento avanza al siguiente subdescriptor, en caso contrario el procesamiento avanza a la etapa S340. Entonces, en la etapa S340, si aún no se ha codificado el número deseado de m grupos de elementos, el procesamiento avanza al grupo de elementos con la siguiente prioridad más alta en el orden de utilización de elementos de grupo, en caso contrario el procesamiento finaliza. Por tanto, las etapas S300, S330 y S340 se refieren al orden en que se realiza el procesamiento y al control de cuántos grupos de elementos codificar, mientras que las etapas S310 y S320 se refieren a la codificación real del descriptor de imágenes local.

El número de grupos m o el número correspondiente de elementos en el descriptor puede almacenarse/transmitirse junto con el descriptor, posiblemente a nivel de imagen.

El hecho de que el orden de utilización de elementos de grupo de la figura 10 sea más económico que el orden de utilización de elementos de la figura 6 da como resultado un codificador más eficaz y económico. Además, tal como se observó anteriormente, para los fines de transcodificación, decodificación y procesamiento, el decodificador también debe conocer el proceso de codificación y el orden de utilización de elementos para poder procesar y comparar descriptores para los fines de las aplicaciones de visión por ordenador relacionadas, lo que significa que el orden de utilización de elementos o bien debe fijarse o bien transmitirse junto con los descriptores. Es posible que diferentes aplicaciones necesiten cambiar el orden de utilización de elementos, posiblemente a nivel de imagen o subimagen, por ejemplo, dando mayor prioridad a los subdescriptores más próximos al centro del descriptor, o dando mayor prioridad a una clase específica de elemento, por ejemplo v_7 en contraposición a v_2 , para lograr un alto rendimiento de reconocimiento con un conjunto de elementos limitado. En este caso, el orden de utilización de elementos debe almacenarse o transmitirse junto con los descriptores. Teniendo en cuenta que los descriptores de

baja tasa de bits normalmente tienen un tamaño de pocos cientos de bytes, el orden de utilización de elementos de grupo de la figura 10 representa una sobrecarga mucho menor que el orden de utilización de elementos de la figura 6. Además, la agrupación de subdescriptores puede ser fija y conocida tanto por el codificador como por el decodificador, o pueden transmitirse junto con los descriptores. Por ejemplo, para la agrupación considerada hasta ahora, el número de grupos y el tamaño de la composición de cada grupo pueden codificarse en menos de 10 bytes.

Como otro ejemplo según una primera realización de la invención, se ilustra una agrupación diferente en la figura 12, donde hay dos grupos, concretamente $g_0 = \{v_0, v_1, v_2, v_3, v_4, v_7, v_8, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}\}$, que contiene todos los subdescriptores periféricos, y $g_1 = \{v_5, v_6, v_9, v_{10}\}$, que contiene los subdescriptores con la menor distancia al centro del descriptor, es decir, todos los subdescriptores centrales. Por tanto, con este ejemplo, el grupo g_1 contiene subdescriptores de diversas distancias al centro del descriptor, pero siempre más lejos del centro que los subdescriptores del grupo g_0 . Dentro de cada grupo, a todos los elementos correspondientes de los subdescriptores del grupo se les asigna la misma prioridad de codificación. Basándose en esta agrupación, puede generarse un orden de utilización de elementos de grupo, que puede adoptar la forma de una lista ordenada de 16 elementos, que puede codificarse en 8 bytes, especificando cada entrada de lista un grupo de subdescriptores y un índice de elemento, tal como se ilustra en la figura 13. El codificador de la figura 11 puede usarse entonces de nuevo para codificar un descriptor según el orden de utilización de elementos de grupo de la figura 13. Cabe señalar que el orden de utilización de elementos de grupo de la figura 13 es simplemente un ejemplo, y pueden generarse diferentes órdenes de utilización de elementos de grupo cambiando las prioridades de las entradas en la lista.

Como otro ejemplo según una primera realización de la invención, se ilustra una agrupación diferente en la figura 14, donde hay cuatro grupos, concretamente $g_0 = \{v_0, v_3, v_{12}, v_{15}\}$, que contiene los subdescriptores con la mayor distancia al centro del descriptor, $g_1 = \{v_1, v_2, v_4, v_7\}$, que contiene un conjunto de cuatro subdescriptores con la segunda mayor distancia al centro del descriptor, $g_2 = \{v_8, v_{11}, v_{13}, v_{14}\}$, que contiene un conjunto de cuatro subdescriptores diferentes de nuevo con la segunda mayor distancia al centro del descriptor, y $g_3 = \{v_5, v_6, v_9, v_{10}\}$, que contiene los subdescriptores con la menor distancia al centro del descriptor. Por tanto, con este ejemplo, los subdescriptores de los grupos g_1 y g_2 están a la misma distancia del centro del descriptor. Esta agrupación se obtiene a partir de la agrupación de la figura 9 subdividiendo el grupo original g_1 en los nuevos grupos g_1 y g_2 . La ventaja de esto es que da como resultado grupos con el mismo número de subdescriptores, lo que es deseable en implementaciones de codificador optimizadas. Dentro de cada grupo, a todos los elementos correspondientes de los subdescriptores del grupo se les asigna la misma prioridad de codificación. Basándose en esta agrupación, puede generarse un orden de utilización de elementos de grupo, que puede adoptar la forma de una lista ordenada de 32 elementos, que puede codificarse en 20 bytes, especificando cada entrada de lista un grupo de subdescriptores y un índice de elemento, tal como se ilustra en la figura 15. El codificador de la figura 11 puede usarse entonces de nuevo para codificar un descriptor según el orden de utilización de elementos de grupo de la figura 15. Cabe señalar que el orden de utilización de elementos de grupo de la figura 15 es simplemente un ejemplo, y pueden generarse diferentes órdenes de utilización de elementos de grupo cambiando las prioridades de las entradas en la lista.

Claramente, la agrupación de la figura 14 en cuatro grupos no es única. Por ejemplo, en la figura 16 se muestra una agrupación alternativa, que contiene los grupos $g_0 = \{v_0, v_3, v_{12}, v_{15}\}$, $g_1 = \{v_1, v_2, v_{13}, v_{14}\}$, $g_2 = \{v_4, v_7, v_8, v_{11}\}$, y $g_3 = \{v_5, v_6, v_9, v_{10}\}$. Por tanto, en la figura 16, los grupos g_0 y g_3 son idénticos a los de la figura 14, pero los subdescriptores de g_1 y g_2 se han intercambiado en comparación con la figura 14 de modo que cada uno de esos dos grupos contiene un subdescriptor de la parte superior izquierda, la parte superior derecha, la parte inferior izquierda y la parte inferior derecha de la cuadrícula de subdescriptores.

Una diferencia entre el codificador de la figura 7 y el codificador de la figura 11 es que, a diferencia del primero que produce descriptores de cualquier longitud, el segundo permite la codificación de descriptores cuya longitud tiene una granularidad decidida por los grupos de subdescriptores. En la práctica, el segundo puede estar configurado para producir descriptores de cualquier longitud, tal como se demostrará más adelante.

Segunda realización

En una segunda realización de la invención, los subdescriptores se agrupan en primer lugar según su distancia desde el centro (primera condición) y en segundo lugar según sus distancias recíprocas (segundo condición).

En este contexto, la distancia entre subdescriptores puede adoptar de nuevo la forma de una distancia euclidiana u otras medidas de distancia adecuadas, tales como la distancia de Manhattan.

La segunda condición puede ser, por ejemplo, que un grupo no contenga subdescriptores cuya distancia entre sí esté por debajo de un umbral predeterminado. Por tanto, puede establecerse el umbral predeterminado para evitar, por ejemplo, la agrupación de subdescriptores vecinos.

El objetivo de la condición secundaria es garantizar que los subdescriptores de los diferentes grupos, siempre que sea posible, se extraigan de posiciones relativamente distantes de la cuadrícula de subdescriptores, aumentando de ese modo el contenido de información de los descriptores de muy pocos números de características. Cabe señalar que la condición secundaria no siempre se satisface, por ejemplo no es posible satisfacerla para un grupo que

contiene todos los subdescriptores centrales v_5, v_6, v_9 y v_{10} .

Por ejemplo, una agrupación de este tipo se ilustra en la figura 17, donde hay cuatro grupos, concretamente $g_0 = \{v_0, v_3, v_{12}, v_{15}\}$, que contiene los subdescriptores con la mayor distancia al centro del descriptor, $g_1 = \{v_1, v_7, v_8, v_{14}\}$, que contiene un conjunto de cuatro subdescriptores con la segunda mayor distancia al centro del descriptor, $g_2 = \{v_2, v_4, v_{11}, v_{13}\}$, que contiene un conjunto de cuatro subdescriptores diferentes de nuevo con la segunda mayor distancia al centro del descriptor, y $g_3 = \{v_5, v_6, v_9, v_{10}\}$, que contiene los subdescriptores con la menor distancia al centro del descriptor. Por tanto, con este ejemplo, los subdescriptores de los grupos g_0, g_1 y g_2 satisfacen la condición de que no contienen subdescriptores vecinos. Dentro de cada grupo, a todos los elementos correspondientes de los subdescriptores del grupo se les asigna la misma prioridad de codificación. Basándose en esta agrupación, puede generarse un orden de utilización de elementos de grupo tal como el ilustrado en la figura 15, y puede usarse entonces de nuevo un codificador tal como el ilustrado en la figura 11 para codificar un descriptor según el orden de utilización de elementos de grupo de la figura 15.

Claramente, también pueden emplearse condiciones alternativas basándose en la distancia de subdescriptores dentro de un grupo, tal como maximizando la distancia total entre subdescriptores, etc.

Tercera realización

En una tercera realización de la invención, los subdescriptores se agrupan según su distancia desde el centro y los subdescriptores de cada grupo se codifican en una secuencia definida según sus características de codificación correspondientes, tal como sus transformadas correspondientes.

Por ejemplo, considerando la agrupación de la figura 17 de la segunda realización de la invención conjuntamente con el patrón de despliegue de transformadas de la figura 3, puede observarse que, dentro de cada grupo, se transforman dos subdescriptores según la transformada A y se transforman dos subdescriptores según la transformada B. Por tanto, es posible establecer una condición de secuencia de codificación común en que, para cada grupo, la secuencia de utilización de transformadas debe ser "A A B B", es decir, una secuencia de codificación mediante la cual, para cada grupo de subdescriptores, la transformada del primer subdescriptor en el grupo es la transformada A, la transformada del segundo subdescriptor en el grupo también es la transformada A, la transformada del tercer subdescriptor en el grupo es la transformada B, y la transformada del cuarto subdescriptor en el grupo también es la transformada B. Por tanto, hay grupos $g_0 = \{v_0, v_{15}, v_3, v_{12}\}$, que contiene los subdescriptores con la mayor distancia al centro del descriptor, $g_1 = \{v_7, v_8, v_1, v_{14}\}$, que contiene un conjunto de cuatro subdescriptores con la segunda mayor distancia al centro del descriptor, $g_2 = \{v_2, v_{13}, v_4, v_{11}\}$, que contiene un conjunto de cuatro subdescriptores diferentes de nuevo con la segunda mayor distancia al centro del descriptor, y $g_3 = \{v_5, v_{10}, v_6, v_9\}$, que contiene los subdescriptores con la menor distancia al centro del descriptor. Dentro de cada grupo, a todos los elementos correspondientes de los subdescriptores del grupo se les asigna la misma prioridad de codificación. Además, dentro de cada grupo, la información de utilización de transformada para los cuatro subdescriptores en el grupo siempre es "A A B B", lo que significa que una implementación de codificador eficaz no necesita identificar la transformada aplicable de cada subdescriptor.

La figura 18 ilustra el funcionamiento de un codificador de este tipo que usa el orden de utilización de elementos de grupo de la figura 15 y que está configurado para codificar los m grupos de elementos superiores en dicho orden de utilización de elementos de grupo. Con el codificador de la figura 18, la codificación del descriptor comienza con el grupo de elementos con la máxima prioridad (prioridad 1) en el orden de utilización de elementos de grupo, es decir el elemento v_0 del grupo g_3 que comprende los subdescriptores v_5, v_{10}, v_6 y v_9 . En la etapa S400, la codificación del descriptor comienza en el primer subdescriptor del grupo, es decir v_5 . Dado que todos los grupos tienen un orden de utilización de transformada común y fijo, el procesamiento avanza a la etapa S410 donde tiene lugar la codificación del elemento, es decir $v_{5,0}$. En la etapa S420, si el subdescriptor actual no es el último subdescriptor en el grupo, el procesamiento avanza al siguiente subdescriptor, en caso contrario el procesamiento avanza a la etapa S430. Entonces, en la etapa S430, si aún no se ha codificado el número deseado de m grupos de elementos, el procesamiento avanza al grupo de elementos con la siguiente prioridad más alta en el orden de utilización de elementos de grupo, en caso contrario el procesamiento finaliza. Por tanto, las etapas S400, S420 y S430 se refieren al orden en que se realiza el procesamiento y al control de cuántos grupos de elementos codificar, mientras que sólo la etapa S410 se refiere a la codificación real del descriptor de imágenes local.

En la descripción anterior, la condición de secuencia de codificación común se define en las transformadas que se utilizan dentro de cada grupo, pero esta condición también puede definirse en otras características de codificación, tales como el tipo y nivel de cuantificación, o combinaciones de los mismos.

Claramente, si se requiere, pueden aplicarse diferentes condiciones de secuencia de codificación a diferentes grupos de subdescriptores. Por ejemplo, en la primera realización de la invención, la agrupación según la figura 9 da como resultado tres grupos, concretamente g_0 , que contiene los cuatro subdescriptores con la mayor distancia al centro del descriptor, g_1 , que contiene los ocho subdescriptores con la segunda mayor distancia al centro del descriptor, y g_2 , que contiene los cuatro subdescriptores con la menor distancia al centro del descriptor. En este caso, debido al diferente tamaño del grupo, puede aplicarse a los grupos g_0 y g_2 una condición de secuencia de

codificación de que la secuencia de utilización de transformada debe ser "A A B B", y puede aplicarse al grupo g_1 una condición de secuencia de codificación diferente de que la secuencia de utilización de transformada debe ser "A A A B B B".

5 Cuarta realización

En una cuarta realización de la invención, los subdescriptores se agrupan según su distancia desde el centro y los subdescriptores de cada grupo se ordenan según su distancias entre ellos.

10 La condición de ordenación puede ser, por ejemplo, que la distancia entre subdescriptores consecutivos dentro de un grupo no debe estar por debajo de un umbral predeterminado. Por tanto, puede establecerse el umbral predeterminado para evitar, por ejemplo, subdescriptores vecinos consecutivos dentro de un grupo.

15 Una condición de orden diferente puede ser, por ejemplo, que la distancia entre subdescriptores consecutivos dentro de un grupo se maximice.

Esta ordenación es particularmente útil cuando se emplean grandes grupos de subdescriptores, para reducir el tamaño del orden de utilización de elementos de grupo y la codificación de grupo parcial, tal como se explica a continuación. Cabe señalar que este tipo de ordenación no siempre es posible, por ejemplo no es posible no tener
20 descriptores vecinos consecutivos para un grupo que contiene todos los subdescriptores centrales v_5, v_6, v_9 y v_{10} .

Por ejemplo, considerando la agrupación ilustrada en la figura 9, puede observarse que los subdescriptores dentro de cada grupo pueden ordenarse para maximizar la distancia entre subdescriptores consecutivos, comenzando desde el subdescriptor con el índice más bajo dentro del grupo, dando como resultado los grupos $g_0 = \{v_0, v_{15}, v_3, v_{12}\}$, $g_1 = \{v_1, v_{14}, v_2, v_{13}, v_7, v_8, v_{11}, v_4\}$ y $g_2 = \{v_5, v_{10}, v_6, v_9\}$. Dentro de cada grupo, a todos los elementos correspondientes de los subdescriptores del grupo se les asigna la misma prioridad de codificación.
25

Basándose en esta agrupación y ordenación de subdescriptores dentro de cada grupo, puede generarse un orden de utilización de elementos de grupo tal como el ilustrado en la figura 10, y puede usarse entonces de nuevo un codificador tal como el ilustrado en la figura 11 para codificar un descriptor según el orden de utilización de elementos de grupo de la figura 10. Alternativamente, puede emplearse un codificador tal como el ilustrado en la figura 19.
30

Más específicamente, una diferencia entre el codificador de la figura 7 y los codificadores de la figura 11 y la figura 18 es que, a diferencia del primero que produce descriptores de cualquier longitud, el segundo permite la codificación de descriptores cuya longitud tiene una granularidad decidida por los grupos de subdescriptores. En la práctica, sin embargo, cada uno de los codificadores de las figuras 11 y 18 puede configurarse fácilmente para codificar descriptores de cualquier longitud mediante la codificación parcial del último grupo de elementos cuando se ha alcanzado la longitud de descriptor deseada. En este contexto, la figura 19 muestra una modificación de este tipo del codificador de la figura 11. En esencia, el codificador de la figura 19 se obtiene intercambiando simplemente las etapas S330 y S340 del codificador de la figura 11, permitiendo que el codificador de la figura 19 termine la codificación de un grupo en cuando se ha codificado el número deseado de elementos. Claramente, también puede aplicarse una modificación análoga al codificador de la figura 18.
35

40 En este contexto, la ordenación de los subdescriptores dentro de un grupo según las distancias entre ellos es sumamente ventajosa puesto que da como resultado elementos consecutivos que se extraen de posiciones relativamente distantes de la cuadrícula de subdescriptores que, en el caso de la codificación parcial de un grupo, aumenta el contenido de información de los descriptores con escaso número de características.

50 Quinta realización

Las realizaciones anteriores de la invención demuestran una codificación eficaz de descriptores de imágenes según un orden de utilización de elementos de grupo que resulta de la agrupación de subdescriptores en grupos de subdescriptores basándose en sus distancias desde el centro del descriptor y/o sus distancias entre sí.
55

En una realización alternativa de la invención, los subdescriptores pueden agruparse de modo que cada subdescriptor dentro de un grupo tenga un conjunto de utilización de elementos idéntico al resto de los subdescriptores dentro de un grupo.

60 Por ejemplo, considérese la agrupación de la figura 20, donde hay cuatro grupos, concretamente $g_0 = \{v_5, v_6, v_9, v_{10}\}$, $g_1 = \{v_1, v_7, v_8, v_{14}\}$, $g_2 = \{v_2, v_4, v_{11}, v_{13}\}$ y $g_3 = \{v_0, v_3, v_{12}, v_{15}\}$. Esta es la misma agrupación que en la figura 17, pero con los índices de grupo reasignados (es decir, g_0 y g_3 se han intercambiado) de modo que los grupos con índices más bajos contengan subdescriptores que estén más cerca del centro del descriptor. La combinación de esta agrupación con los conjuntos de utilización de elementos de las figuras 4a a 4e conduce a los conjuntos de utilización de elementos de grupo de las figuras 21a a 21e.
65

Puesto que el conjunto de elementos utilizados para cada longitud de descriptor debe ser igual que o un subconjunto del conjunto de elementos utilizados para todas las longitudes de descriptor más altas, los conjuntos de utilización de elementos de grupo de las figuras 21a a 21e pueden convertirse en conjuntos de utilización incrementales de elementos de grupo tal como se ilustra en las figuras 22a a 22e donde, para cada longitud de descriptor (por ejemplo DL2 de la figura 22c), el conjunto de utilización de elementos correspondiente muestra sólo los elementos adicionales que constituyen el descriptor de dicha longitud de descriptor, en comparación con la longitud de descriptor inmediatamente inferior (por ejemplo DL1 de la figura 22b).

Basándose en los conjuntos de utilización incrementales de elementos de grupo de las figuras 22a a 22e, un codificador puede generar descriptores que pueden convertirse en longitudes de descriptor más bajas mediante el simple truncamiento del descriptor.

La figura 23 ilustra el funcionamiento de un codificador de este tipo para la codificación de un descriptor de longitud DLk. Más específicamente, en la etapa S600, el procesamiento comienza codificando un descriptor de la longitud de descriptor más baja, es decir, DL0. En la etapa S610, la codificación del descriptor DL0 comienza con la codificación del primer grupo de subdescriptores, es decir, g_0 , y en la etapa S620 la codificación del primer grupo de subdescriptores comienza con la codificación del primer elemento, es decir v_0 . En la etapa S630, si el elemento v_0 no está en uso para g_0 en la longitud de descriptor DL0 según los conjuntos de utilización incrementales de elementos de grupo de la figura 22, el procesamiento avanza a la etapa S680, en caso contrario el procesamiento avanza a la etapa S640. En la etapa S640, se selecciona el primer subdescriptor del grupo g_0 , es decir v_5 , y en la etapa S650 se selecciona la función de transformada apropiada, por ejemplo, según la figura 3. Cabe señalar que el cálculo del descriptor V a partir del descriptor H según dos transformadas diferentes tal como se describe en este caso es sólo un ejemplo. En diferentes realizaciones, el cálculo del descriptor V a partir del descriptor H también puede realizarse según una sola transformada, por ejemplo sólo la transformada A o sólo la transformada B, haciendo innecesaria la etapa S650, o según más de dos transformadas. Entonces, tiene lugar la codificación del elemento $v_{5,0}$ en la etapa 660. En la etapa S670, si el subdescriptor actual no es el último subdescriptor en el grupo, el procesamiento avanza al siguiente subdescriptor en el grupo; en caso contrario, el procesamiento avanza a la etapa S680. En la etapa S680, si el elemento actual no es el último elemento, es decir v_7 , el procesamiento avanza al siguiente elemento, en caso contrario el procesamiento avanza a la etapa S690. En la etapa S690, si el grupo de subdescriptores actual no es el último grupo de subdescriptores, el procesamiento avanza al siguiente grupo de subdescriptores, en caso contrario, el procesamiento avanza a la etapa S695. En la etapa S695, si la longitud de subdescriptor actual no es la longitud de subdescriptor objetivo, el procesamiento avanza a la siguiente longitud de subdescriptor, codificando los elementos adicionales especificados por los conjuntos de utilización incrementales de elementos de grupo. En caso contrario, el procesamiento finaliza.

Aunque los aspectos y las realizaciones de la presente invención se presentan en detalle para el cálculo de descriptores de imágenes robustos, discriminativos, escalables y compactos a partir de un descriptor de imágenes SIFT, la invención es aplicable a otros descriptores de imágenes basándose en histogramas de gradientes, tal como se describe en la solicitud de patente italiana aún no publicada n. TO2012A000602.

A modo de ejemplo únicamente, la figura 24 ilustra un aparato de procesamiento de imágenes conceptual para llevar a cabo el método según la presente invención. Más específicamente, el aparato 110 de procesamiento recibe entrada, que puede comprender datos visuales, tales como datos de imagen o vídeo, descriptores compactos calculados previamente basándose en histogramas de gradientes, descriptores compactos calculados previamente según el método de la presente invención, instrucciones de programación o entrada del usuario, desde el aparato 1000 de entrada, que puede adoptar la forma de un aparato de entrada de usuario, un lector de medios o un receptor de señales transmitidas. El aparato 110 de procesamiento comprende los bloques de procesamiento principales de una unidad 1110 de procesamiento central que controla las operaciones de los otros bloques de procesamiento, una memoria 1120 volátil, una memoria 1130 no volátil, opcionalmente un bloque 1140 extractor de descriptor configurado para generar descriptores basándose en histogramas de gradientes, un bloque 1150 codificador de descriptor compacto configurado para llevar a cabo el método según la presente invención, y opcionalmente un bloque 1160 de procesador de descriptor compacto configurado para procesar dichos descriptores compactos, por ejemplo para establecer o verificar correspondencias visuales. El aparato 110 de procesamiento está conectado al aparato 1900 de salida, que puede adoptar la forma de una unidad de presentación visual, un escritor de medios o un transmisor de señales, que proporciona una salida que puede comprender datos visuales anotados, tales como datos de imagen o vídeo, información de procesamiento tal como correspondencias visuales establecidas o verificadas, o descriptores compactos según el método de la presente invención. Debe entenderse que los bloques de procesamiento y la arquitectura mostrados en la figura 24 son sólo conceptuales y pueden no corresponder exactamente a cada aparato que implementa el método según la invención.

El método para codificar un descriptor de imágenes basándose en un histograma de gradientes y un aparato de procesamiento de imágenes relativo descritos en el presente documento a modo de ejemplo pueden someterse a muchas variaciones posibles sin apartarse del espíritu de novedad de la idea inventiva; también está claro que en la implementación práctica de la invención, los detalles ilustrados pueden tener diferentes formas o pueden reemplazarse por otros elementos técnicamente equivalentes.

Por tanto, puede entenderse fácilmente que la presente invención no se limita a un método para codificar un descriptor de imágenes basándose en un histograma de gradientes y a un aparato de procesamiento de imágenes relativo, sino que pueden someterse a muchas modificaciones, mejoras o reemplazos de partes y elementos equivalentes sin apartarse de la idea inventiva, tal como se especifica claramente en las siguientes reivindicaciones.

5

REIVINDICACIONES

1. Método para codificar un descriptor de imágenes (H) basándose en histogramas de gradiente (h_p) que comprenden cada uno una pluralidad de compartimentos de histograma (hi) y que se calculan en una subregión correspondiente (SR) de una región local (R) alrededor de un punto clave de interés de imagen, usando un descriptor transformado (V) que comprende subdescriptores transformados (v_p) para cada subregión (SR) que se calculan como una transformación del histograma de gradiente (h_p) de subregión correspondiente (SR), en el que cada subdescriptor (v_p) comprende un conjunto de valores (v_j) individualizado cada uno por un índice de elemento (j), caracterizado porque dicho método comprende:
- determinar una pluralidad de grupos (g_x), comprendiendo cada uno un conjunto de subdescriptores transformados (v_p), determinándose el conjunto usando la distancia entre el centro de la subimagen correspondiente (SR) de cada subdescriptor transformado (v_p) y el centro de dicha región local (R), estando cada grupo (g_x) individualizado por un índice de grupo (x);
 - usar una lista de utilización de elementos para indicar la secuencia para codificar los valores (v_j) de los elementos de subdescriptores transformados (v_p), especificando la lista de utilización de elementos una secuencia ordenada de entradas de lista, definiendo cada entrada un par de índices que incluyen dicho índice de grupo (x), que indica un conjunto de subdescriptores transformados, y dicho índice de elemento (j), y la secuencia truncada en una longitud DLn que define un descriptor que tiene una longitud de DLn elementos que es un subconjunto de un descriptor que tiene una longitud de DLp elementos con DLn < DLp, definido por la secuencia de entradas truncada en una longitud DLp mayor que DLn;
 - codificar los valores (v_j) del descriptor transformado (V) según el orden de la lista de utilización de elementos, comprendiendo la secuencia codificada una secuencia de descriptores transformados de diferentes longitudes (DLO, DL1, DL2, DL3, DL4).
2. Método según la reivindicación 1, en el que dentro de cada grupo de subdescriptores (g_x), a todos los valores correspondientes (v_j) de un grupo de subdescriptores (g_x) se les asigna una misma prioridad de codificación en dicha lista de utilización de elementos.
3. Método según la reivindicación 1, en el que dichos subdescriptores (v_p) se agrupan en un primer grupo (g_0) que contiene los subdescriptores correspondientes a subregiones con la mayor distancia central desde el centro de dicha región local (R), un segundo grupo (g_1) que contiene un primer conjunto de subdescriptores correspondientes a subregiones con la segunda mayor distancia central al centro de dicha región local (R), un tercer grupo (g_2) que contiene un segundo conjunto de subdescriptores correspondientes a subregiones con la segunda mayor distancia central al centro de dicha región local (R), y un cuarto grupo (g_3) que contiene los subdescriptores correspondientes a subregiones con la menor distancia central al centro de dicha región local (R).
4. Método según la reivindicación 3, en el que dichos grupos primero, segundo, tercero y cuarto (g_0, g_1, g_2, g_3) contiene el mismo número de subdescriptores.
5. Método según la reivindicación 1, en el que dichos subdescriptores (v_p) se agrupan adicionalmente según las distancias centrales recíprocas de sus subregiones correspondientes y en el que, dentro de cada grupo de subdescriptores (g_x), a todos los valores correspondientes (v_j) de un grupo de subdescriptores (g_x) se les asigna una misma prioridad de codificación en dicha lista de utilización de elementos.
6. Método según la reivindicación 5, en el que dichos subdescriptores (v_p) se agrupan en un primer grupo (g_0) que contiene los subdescriptores correspondientes a subregiones con la mayor distancia central desde el centro de dicha región local (R), un segundo grupo (g_1) que contiene un primer conjunto de subdescriptores correspondientes a subregiones con la segunda mayor distancia central al centro de dicha región local (R), un tercer grupo (g_2) que contiene un segundo conjunto de subdescriptores correspondientes a subregiones con la segunda mayor distancia central al centro de dicha región local (R), y un cuarto grupo (g_3) que contiene los subdescriptores correspondientes a subregiones con la menor distancia central al centro de dicha región local (R).
7. Método según la reivindicación 1, en el que dichos subdescriptores (v_p) se ordenan según las distancias centrales recíprocas de sus subregiones correspondientes y en el que dentro de cada grupo de subdescriptores (g_x), a todos los valores correspondientes (v_j) de un grupo de subdescriptores se les asigna una misma prioridad de codificación en dicha lista de utilización de elementos.
8. Método según la reivindicación 1, en el que los subdescriptores de cada grupo se codifican en una secuencia definida según sus características de codificación correspondientes y en el que, dentro de cada grupo de subdescriptores, a todos los valores correspondientes (v_j) de un grupo de subdescriptores se les asigna una misma prioridad de codificación en dicha lista de utilización de elementos.

9. Método según la reivindicación 8, en el que dichas características de codificación son transformadas correspondientes.
- 5 10. Método según la reivindicación 9, en el que para cada grupo de subdescriptores, la transformada de un primer subdescriptor en el grupo es un primer tipo de transformada (A), la transformada de un segundo subdescriptor en el grupo es dicho primer tipo de transformada (A), la transformada de un tercer subdescriptor en el grupo es un segundo tipo de transformada (B) y la transformada de un cuarto subdescriptor en el grupo es dicho segundo tipo de transformada (B).
- 10 11. Método según la reivindicación 8, en el que dichas características de codificación son el tipo y el nivel de cuantificación, o combinaciones de los mismos.
- 15 12. Método según la reivindicación 1, en el que se da mayor prioridad a una clase específica de valores.
13. Método según la reivindicación 1, en el que se da mayor prioridad a los grupos de subdescriptores más próximos al centro del descriptor.
- 20 14. Método según la reivindicación 1, en el que el número de grupos (m) o el número correspondiente de elementos en el descriptor se almacena o se transmite junto con el descriptor, a nivel de imagen.
15. Método para codificar un descriptor de imágenes (H) según las reivindicaciones 1 a 14, en el que dicho descriptor de imágenes se convierte en un descriptor que tiene una longitud reducida mediante truncamiento de uno o más de los últimos elementos de dicho descriptor.
- 25 16. Método para codificar un descriptor de imágenes según la reivindicación 1, en el que dicho descriptor transformado (V) adopta la forma de una secuencia de 20 ó 40 ó 64 u 80 ó 128 elementos.
- 30 17. Aparato de procesamiento de imágenes que comprende medios para llevar a cabo el método según una cualquiera de las reivindicaciones 1 a 16.

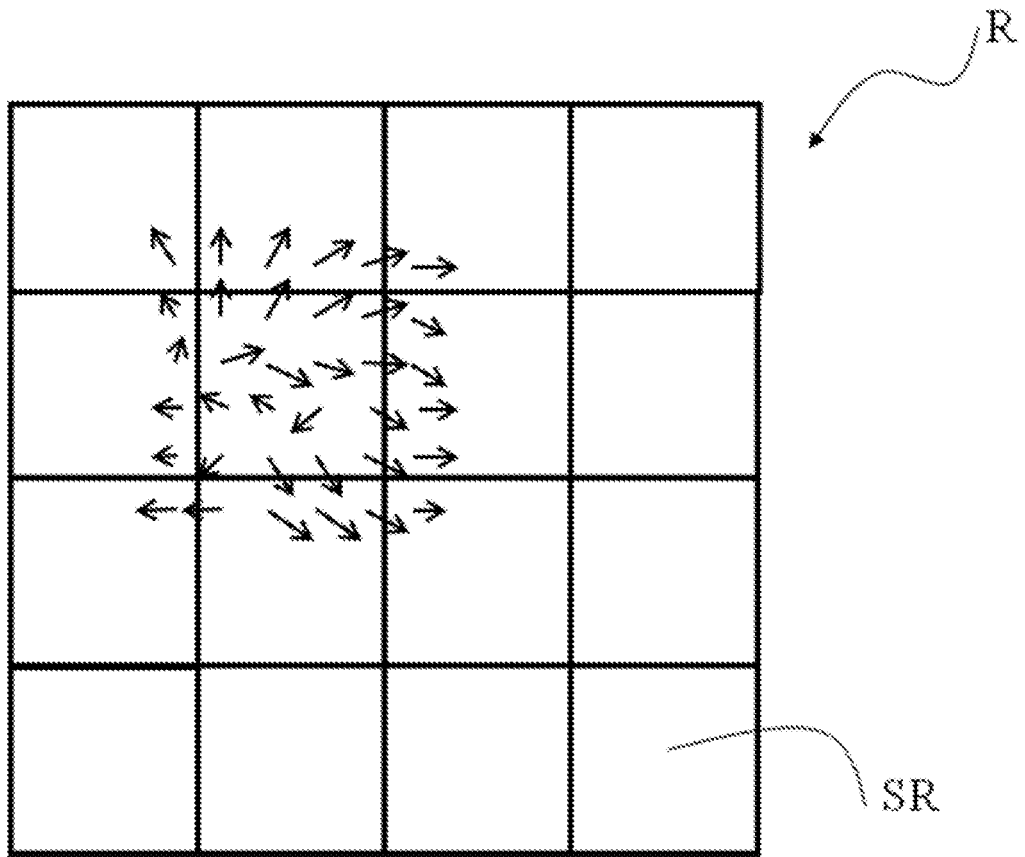


Fig. 1a

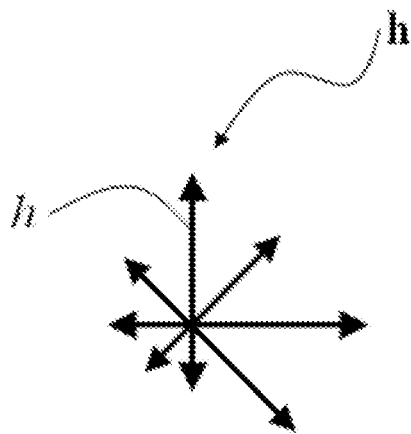


Fig. 1b

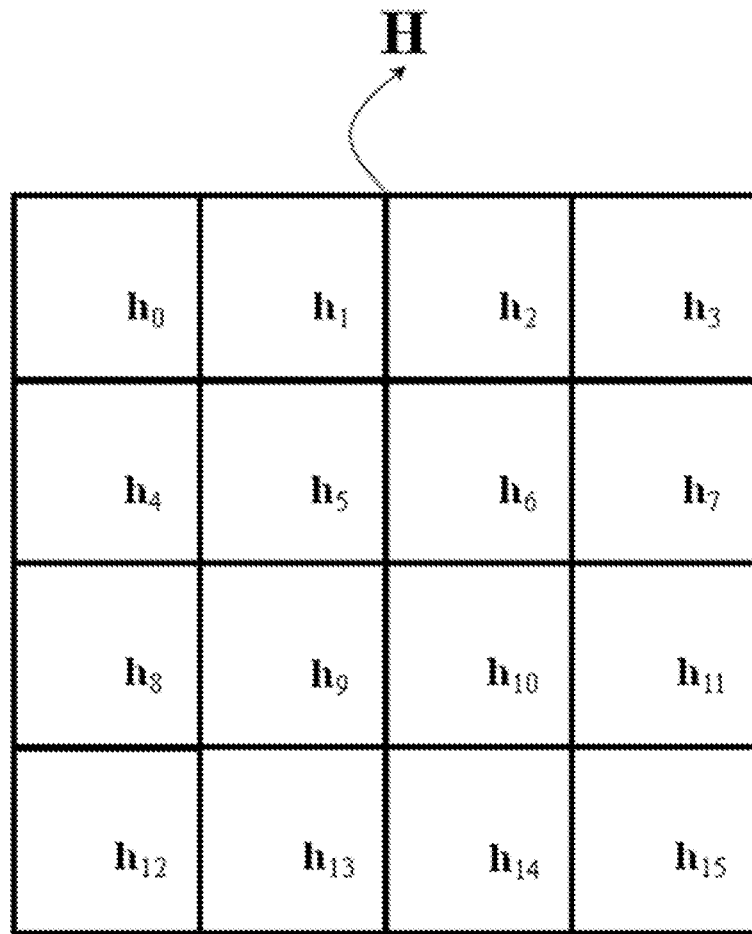


Fig. 2a

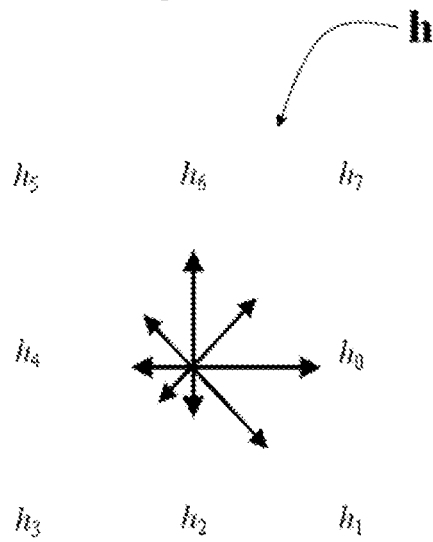


Fig. 2b

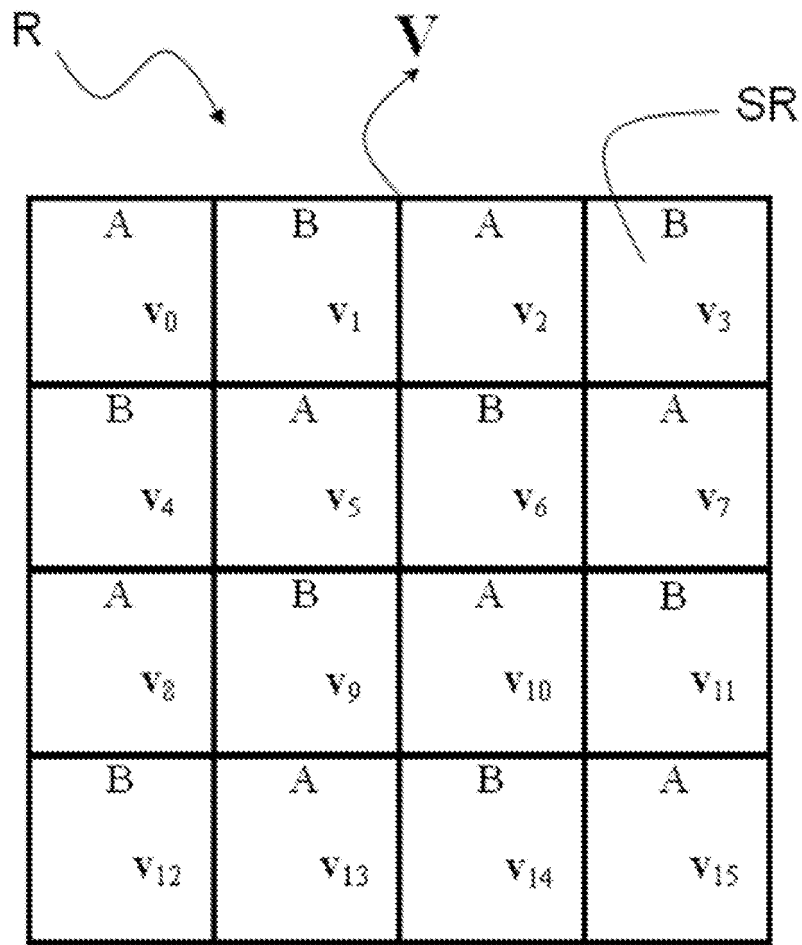


Fig. 3

v_0				v_1				v_2				v_3			
v_0				v_0				v_0				v_0			
v_4				v_5				v_6				v_7			
v_0				v_0			v_6	v_0			v_6	v_0			
v_8				v_9				v_{10}				v_{11}			
v_0				v_0			v_6	v_0			v_6	v_0			
v_{12}				v_{13}				v_{14}				v_{15}			
v_0				v_0				v_0				v_0			

Fig. 4a

v_0				v_1				v_2				v_3			
v_0	v_1			v_0	v_1			v_0	v_1			v_0	v_1		
v_4				v_5				v_6				v_7			
v_0	v_1			v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1		
v_8				v_9				v_{10}				v_{11}			
v_0	v_1			v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1		
v_{12}				v_{13}				v_{14}				v_{15}			
v_0	v_1			v_0	v_1			v_0	v_1			v_0	v_1		

Fig. 4b

v_0				v_1				v_2				v_3			
v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2	
v_4				v_5				v_6				v_7			
v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2	
v_8				v_9				v_{10}				v_{11}			
v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2	
v_{12}				v_{13}				v_{14}				v_{15}			
v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2	

Fig. 4c

v_0				v_1				v_2				v_3			
v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2	
v_4				v_5				v_6				v_7			
v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2	
v_8				v_9				v_{10}				v_{11}			
v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2	
v_{12}				v_{13}				v_{14}				v_{15}			
v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2		v_0	v_1	v_2	

Fig. 4d

v_0				v_1				v_2				v_3			
v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_4				v_5				v_6				v_7			
v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_8				v_9				v_{10}				v_{11}			
v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_{12}				v_{13}				v_{14}				v_{15}			
v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7

Fig. 4e

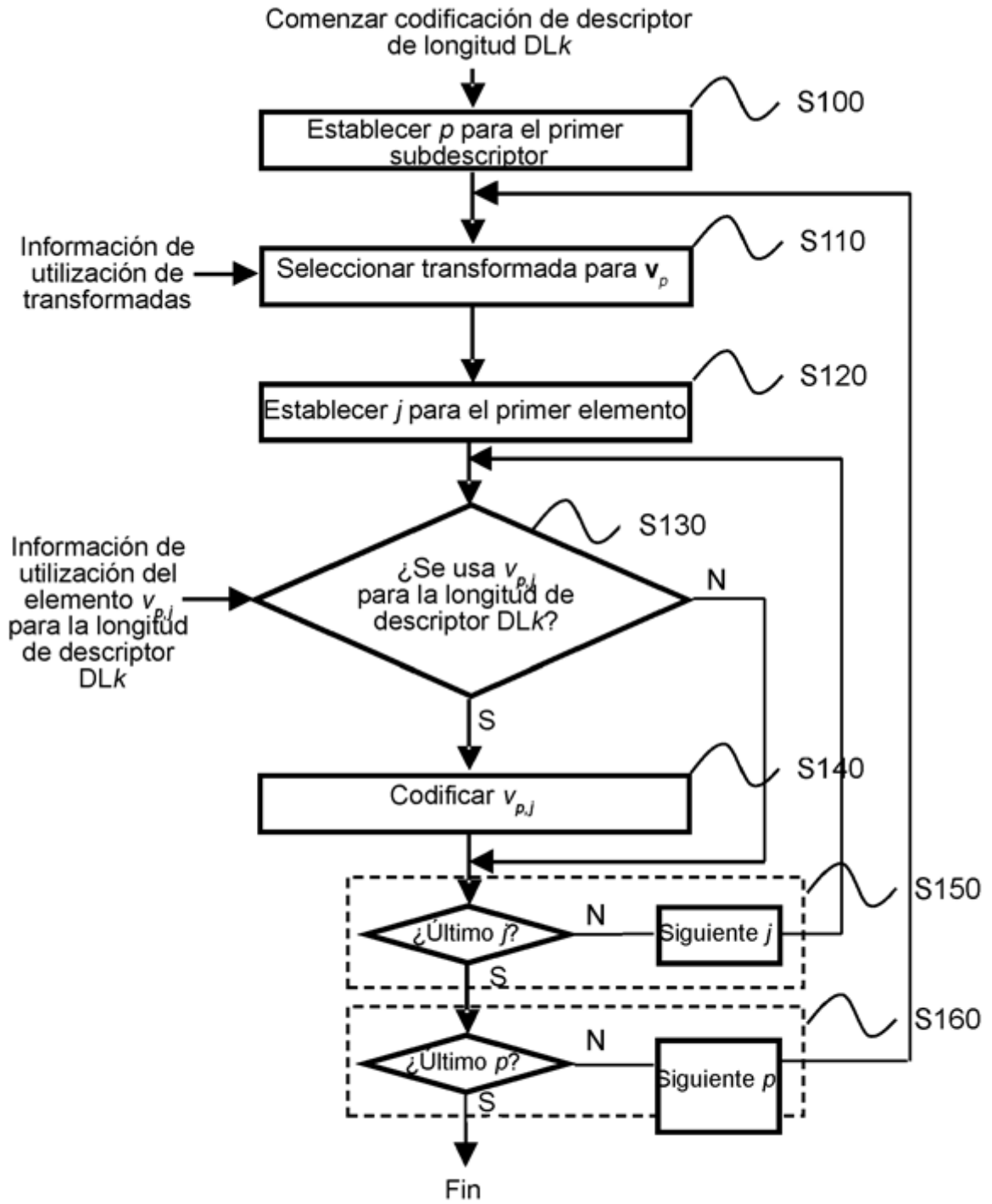


Fig. 5

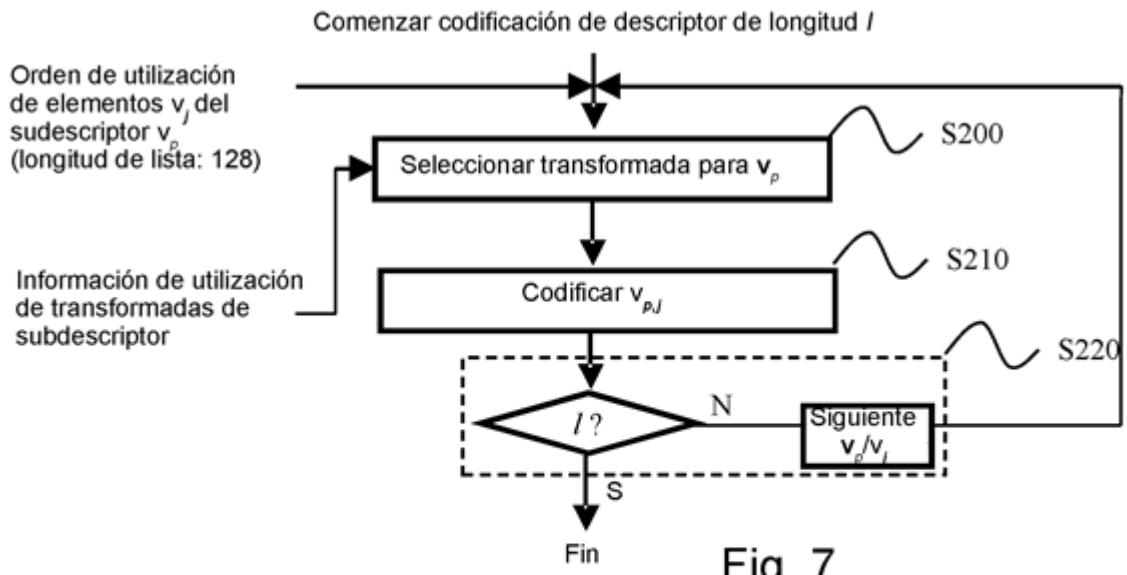


Fig. 7

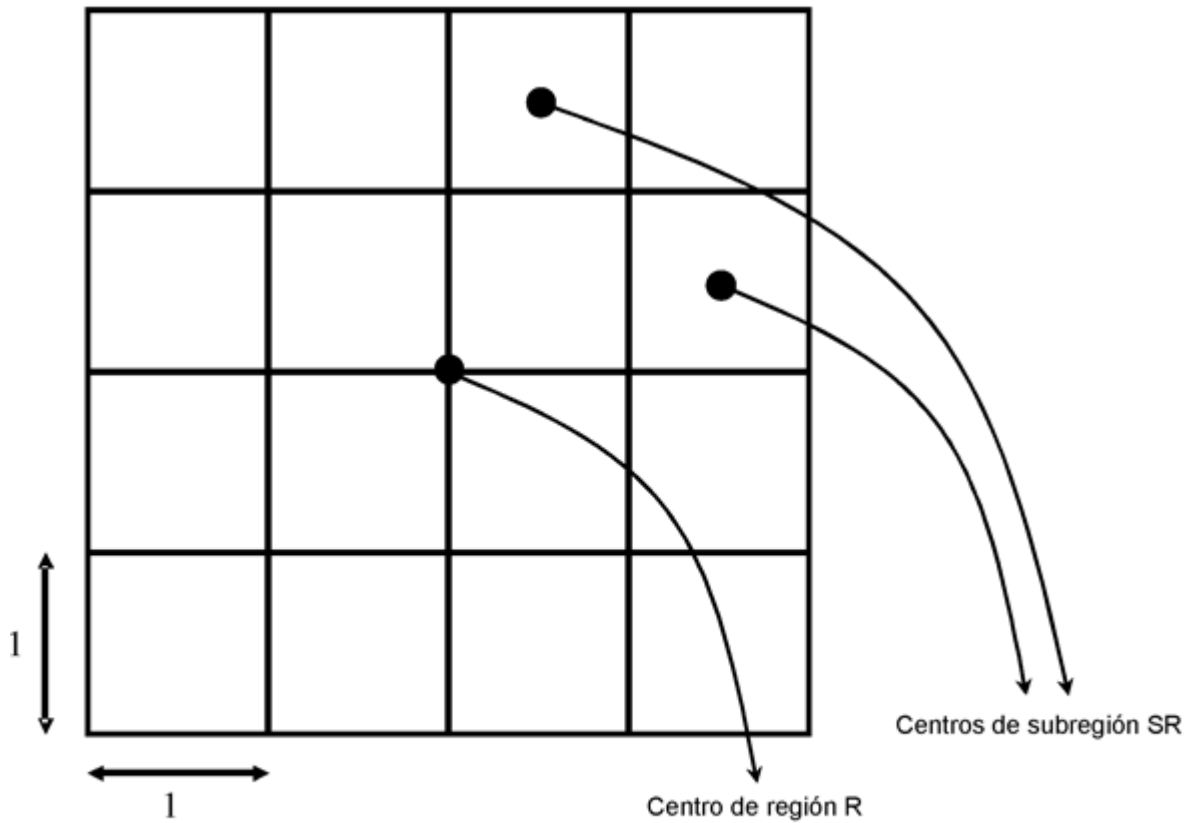


Fig. 8

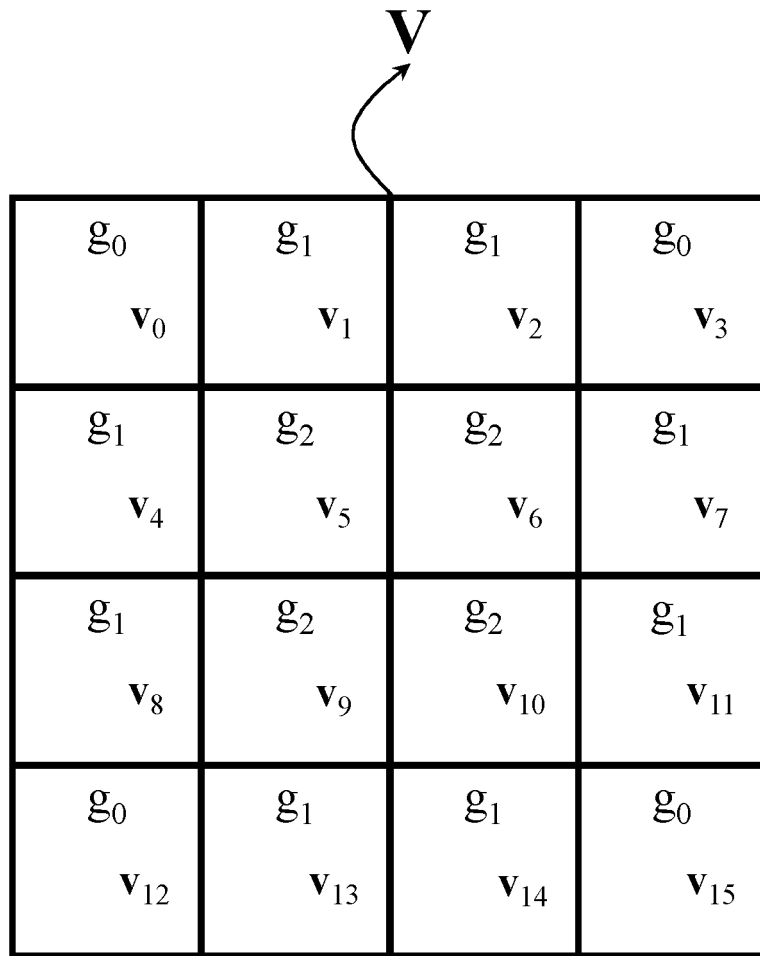


Fig. 9

Prioridad	Grupo g_x	Elemento v_j
1	2	0
2	1	0
3	0	0
4	2	6
5	2	1
6	1	1
7	0	1
8	2	2
9	1	2
10	0	2
11	1	6
12	0	6
13	2	7
14	1	7
15	0	7
16	2	3
17	1	3
18	0	3
19	2	4
20	1	4
21	0	4
22	2	5
23	1	5
24	0	5

Fig. 10

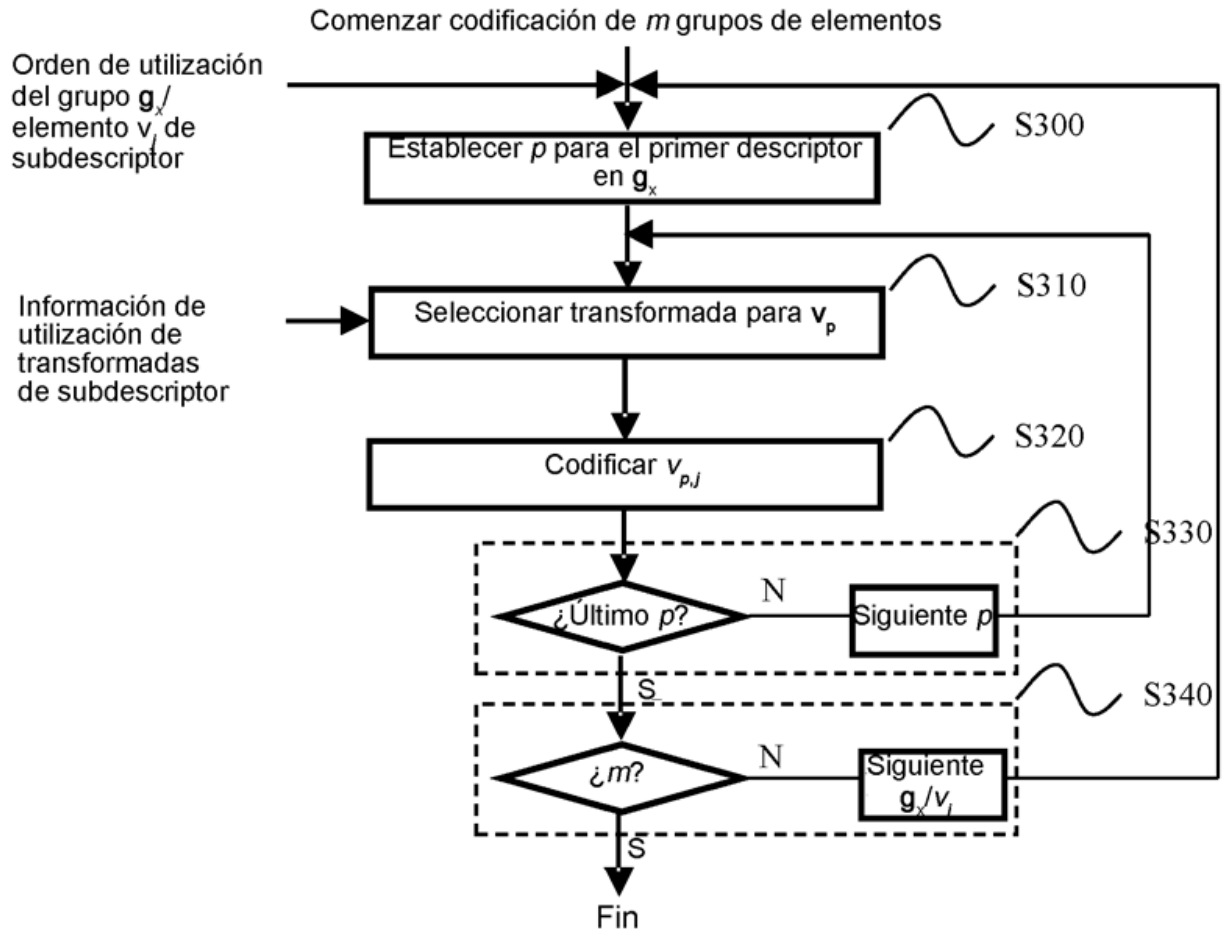


Fig. 11

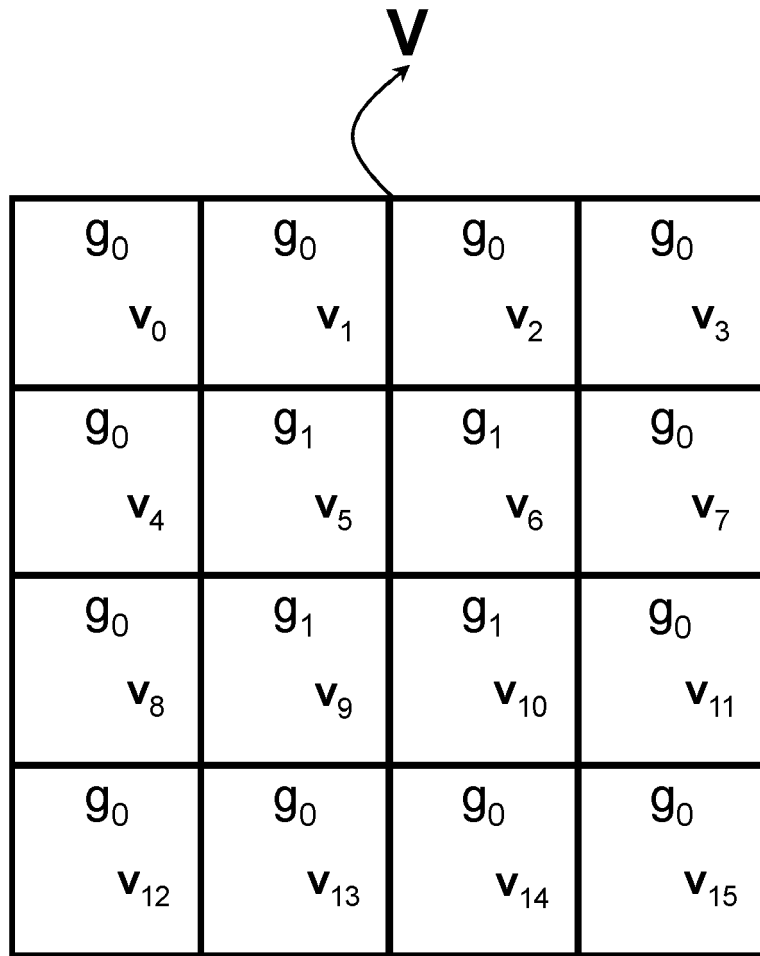


Fig. 12

Prioridad	Grupo g_x	Elemento v_j
1	1	0
2	0	0
3	1	6
4	1	1
5	0	1
6	1	2
7	0	2
8	0	6
9	1	7
10	0	7
11	1	3
12	0	3
13	1	4
14	0	4
15	1	5
16	0	5

Fig. 13

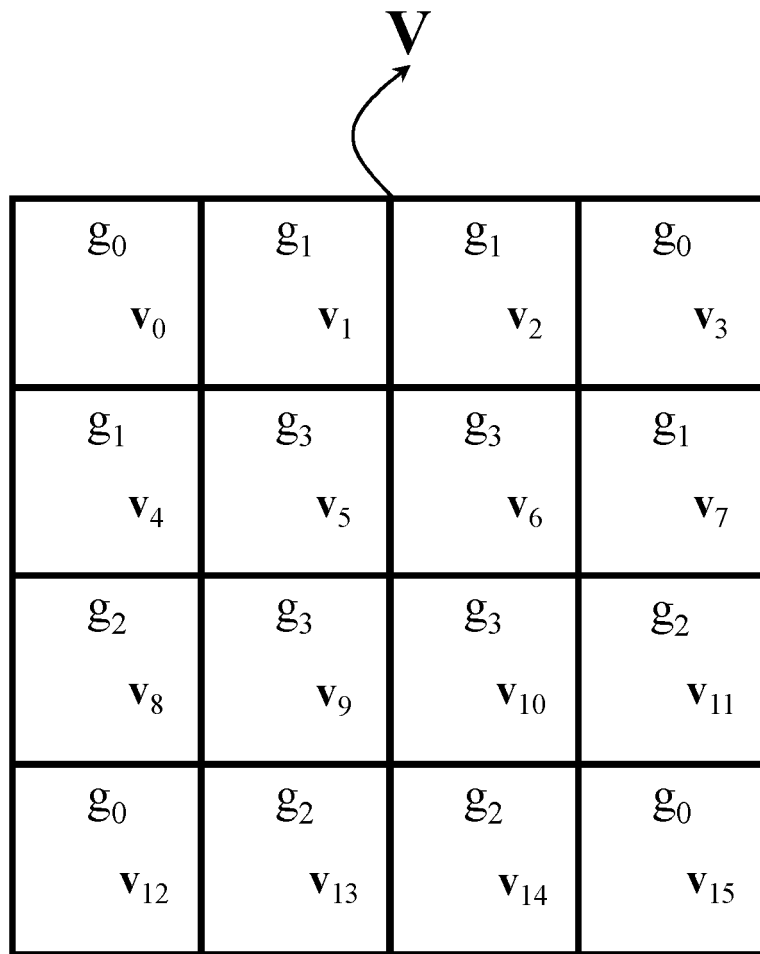


Fig. 14

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

Fig. 15

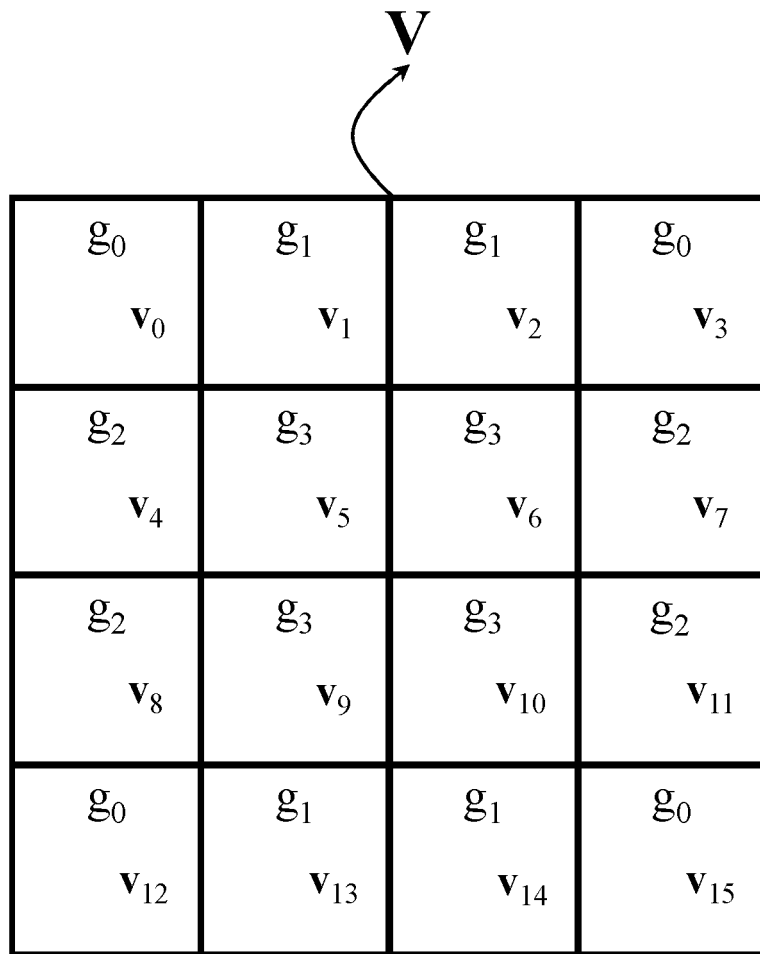


Fig. 16

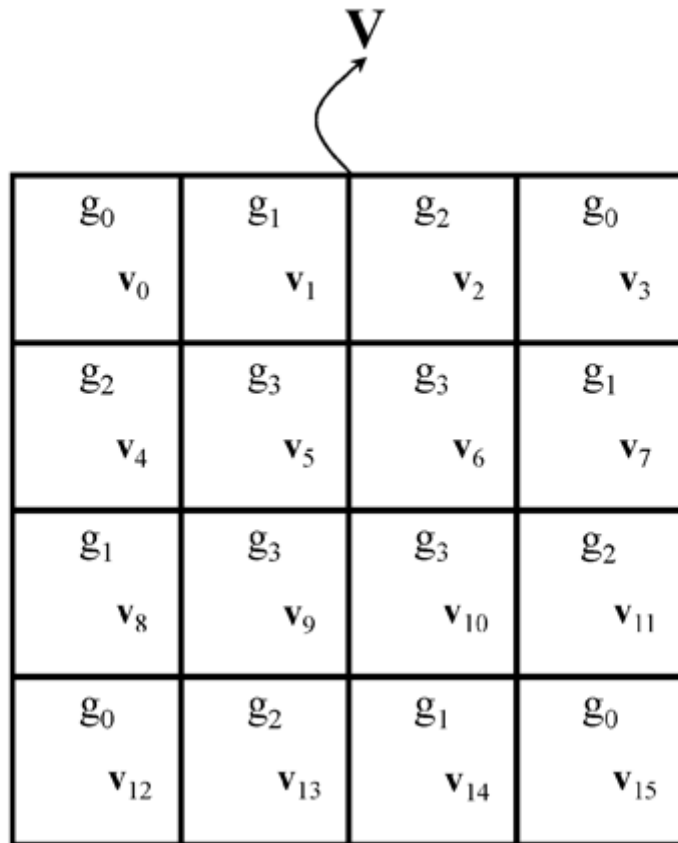


Fig. 17

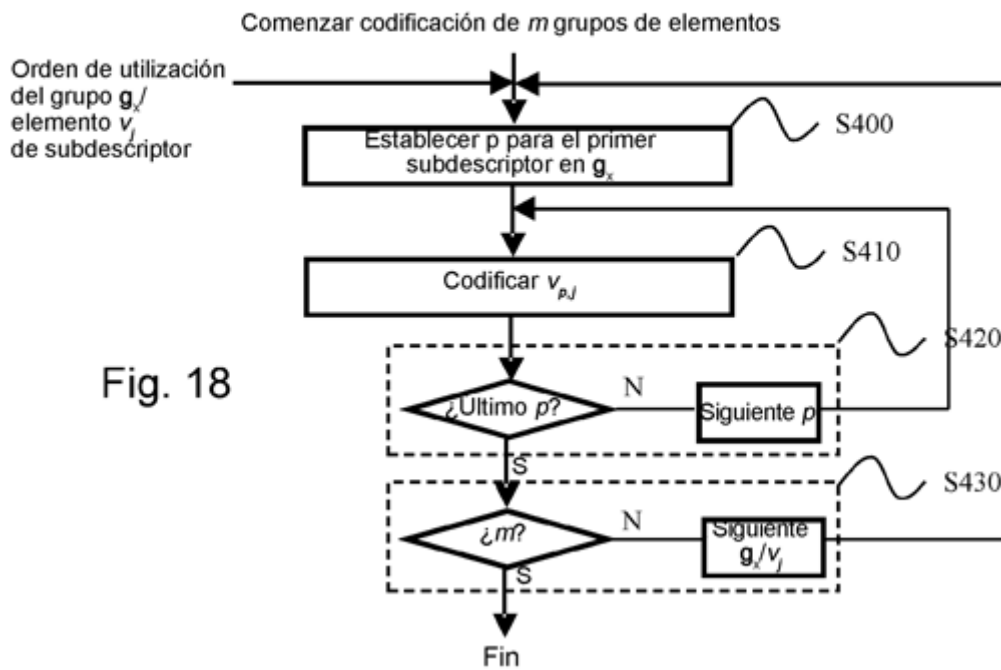
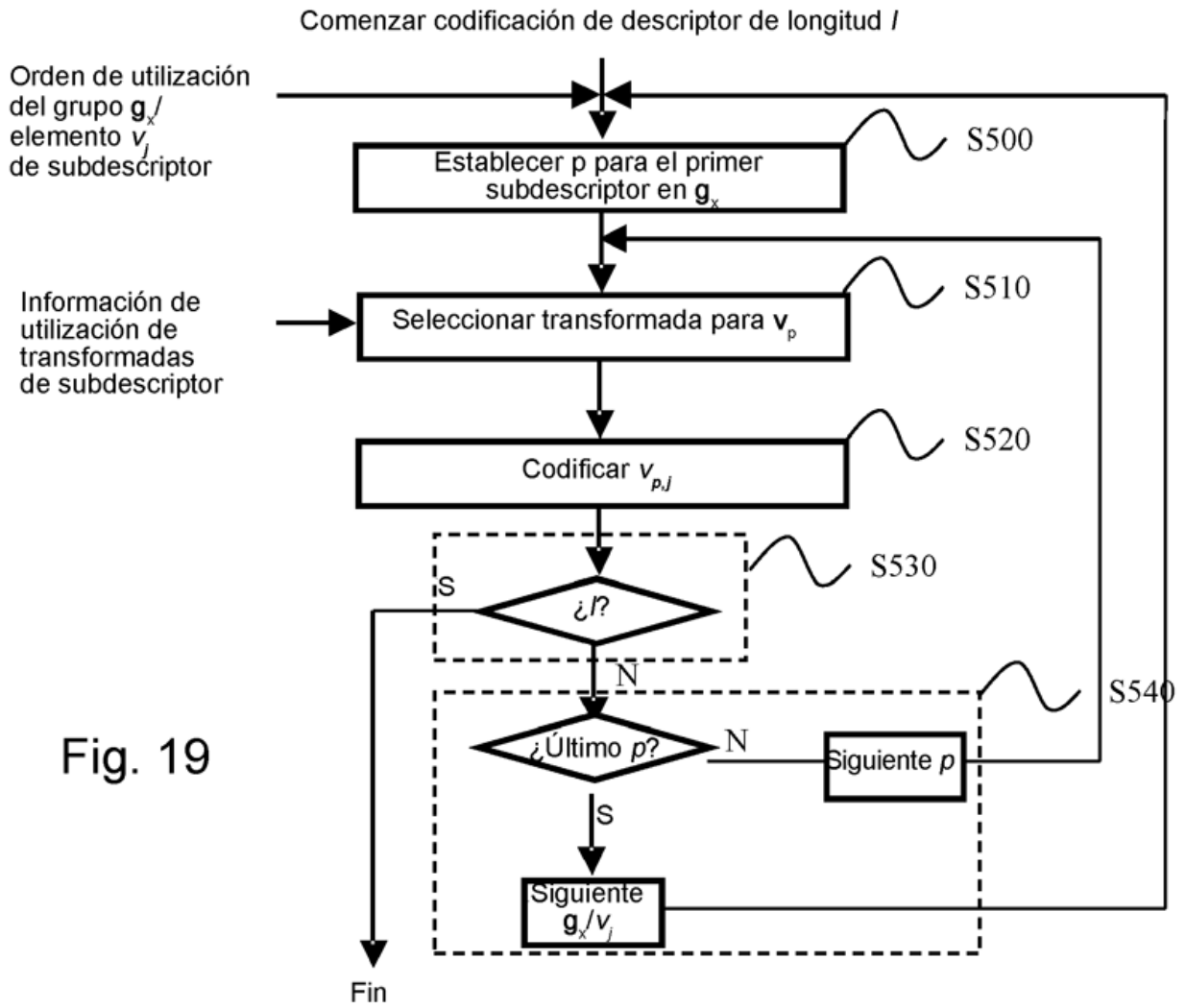


Fig. 18



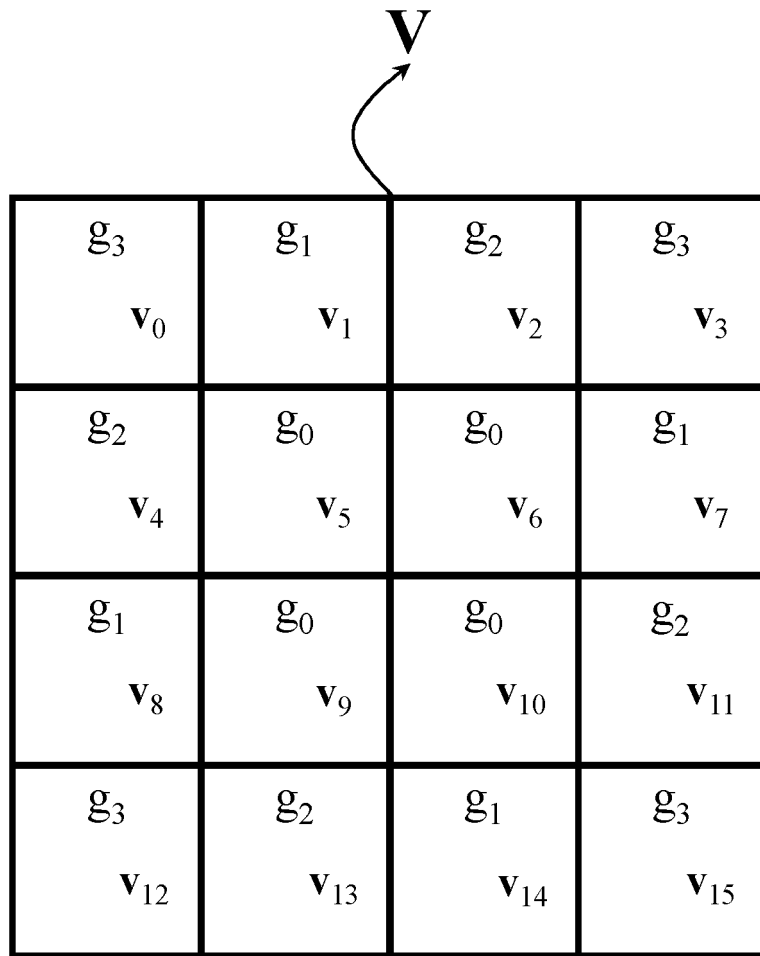


Fig. 20

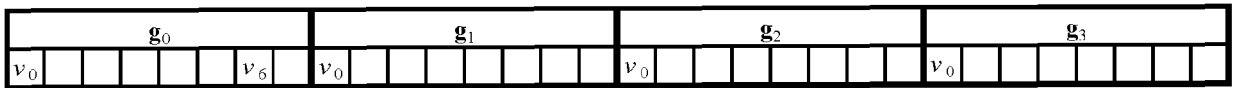


Fig. 21a

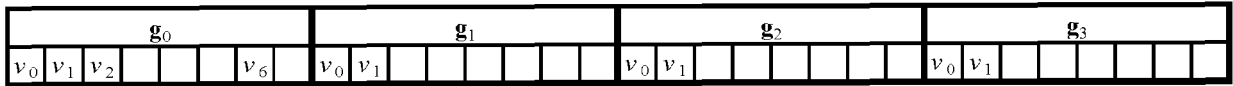


Fig. 21b

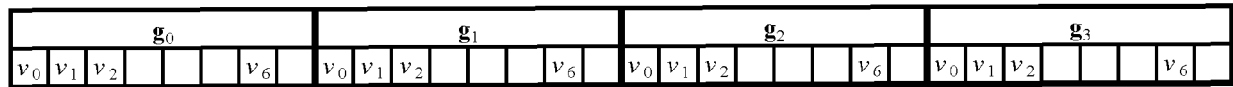


Fig. 21c

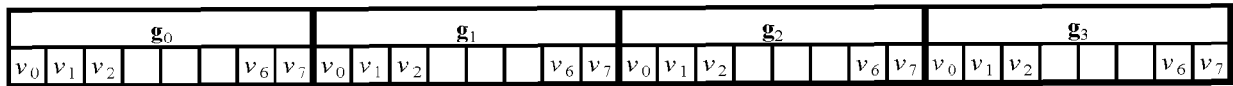


Fig. 21d

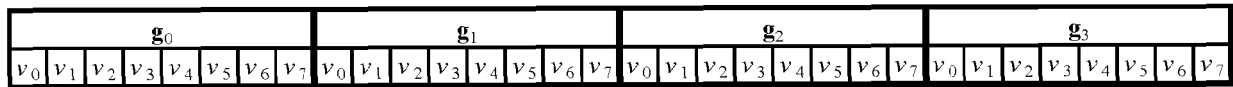


Fig. 21e

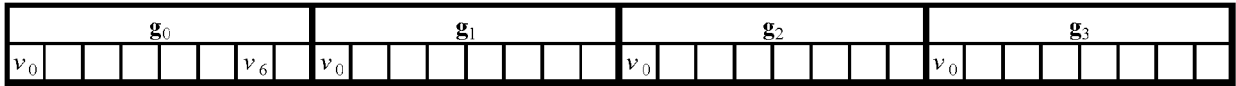


Fig. 22a

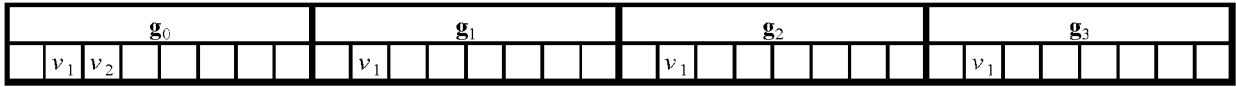


Fig. 22b

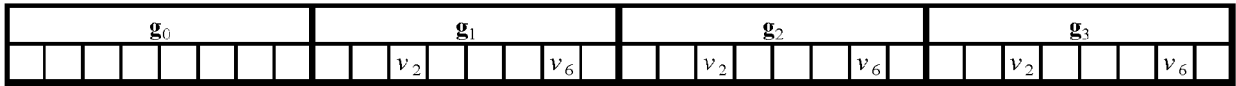


Fig. 22c

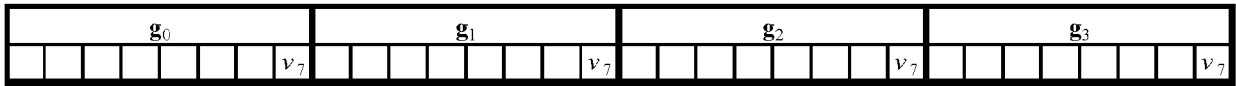


Fig. 22d

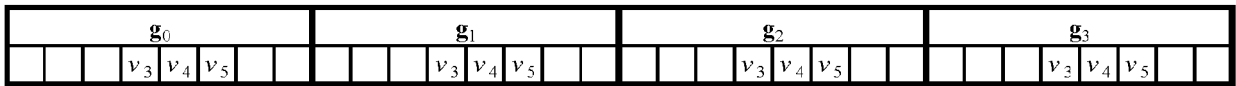


Fig. 22e

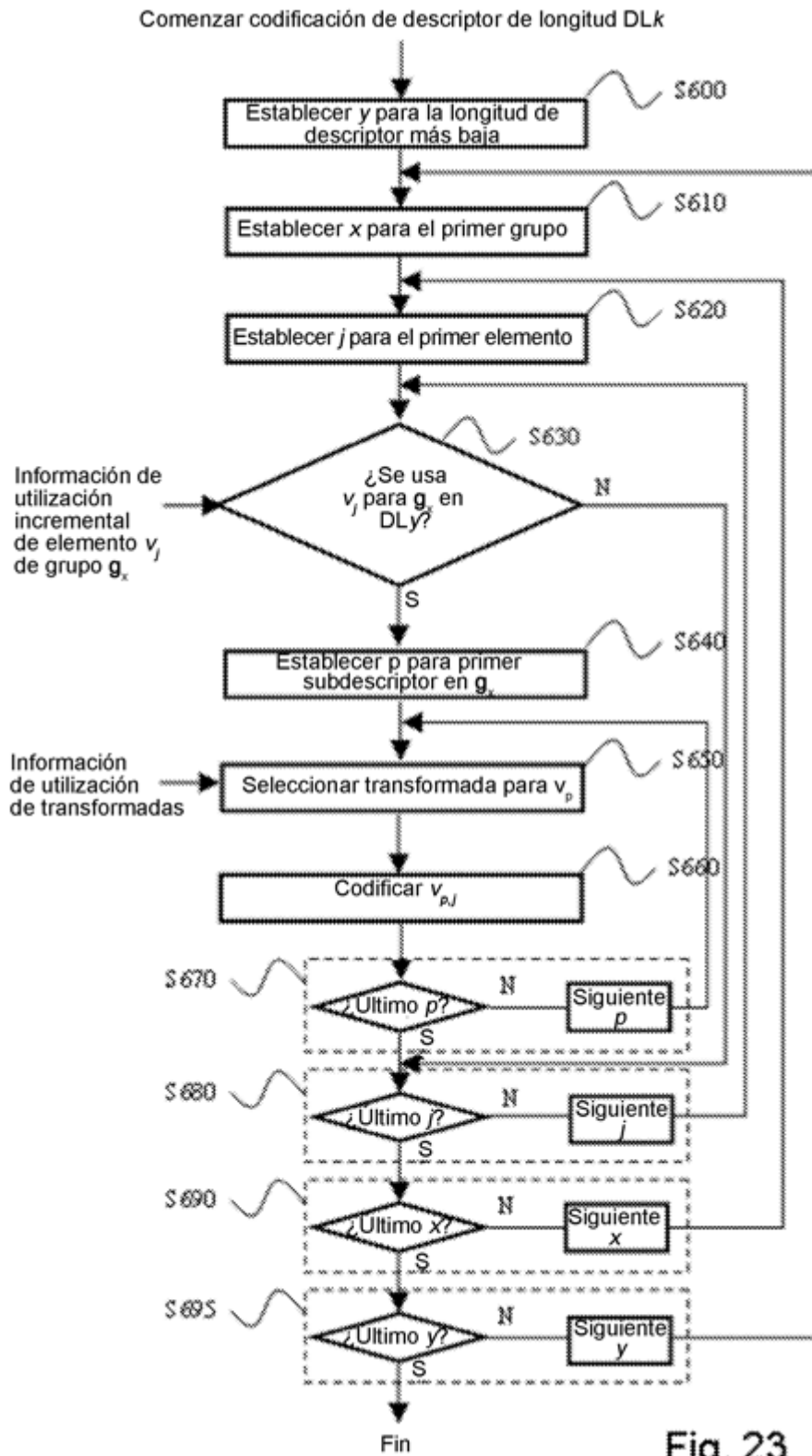


Fig. 23

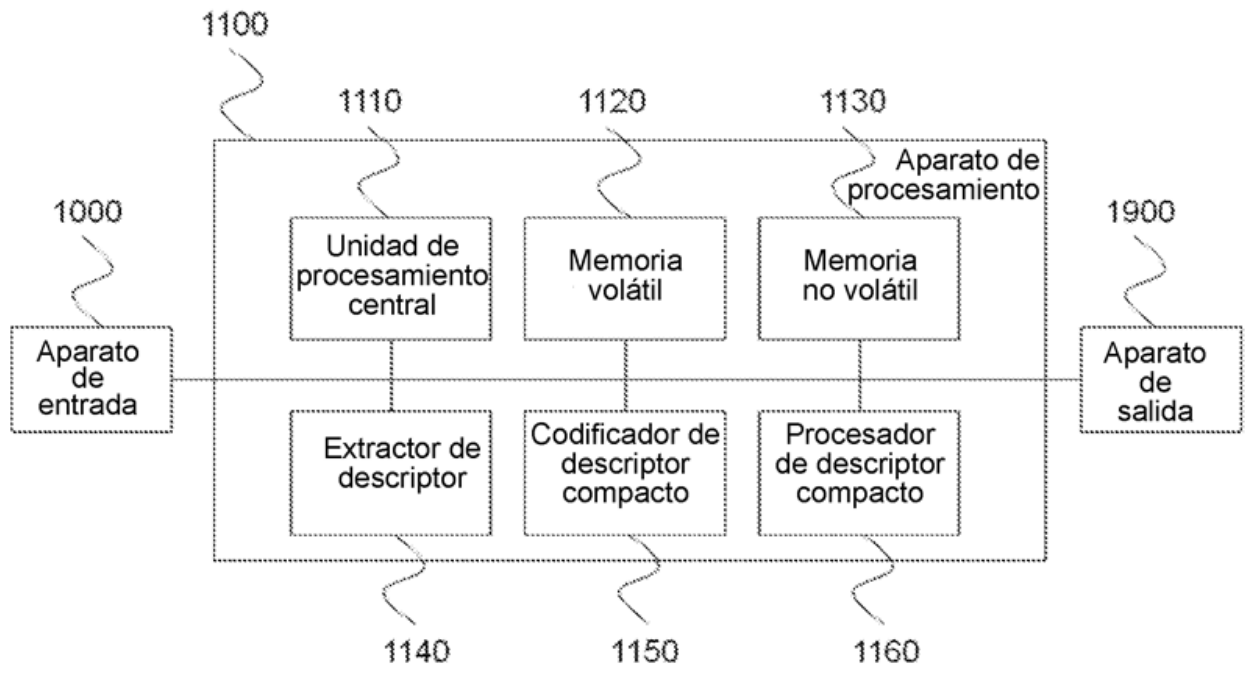


Fig. 24