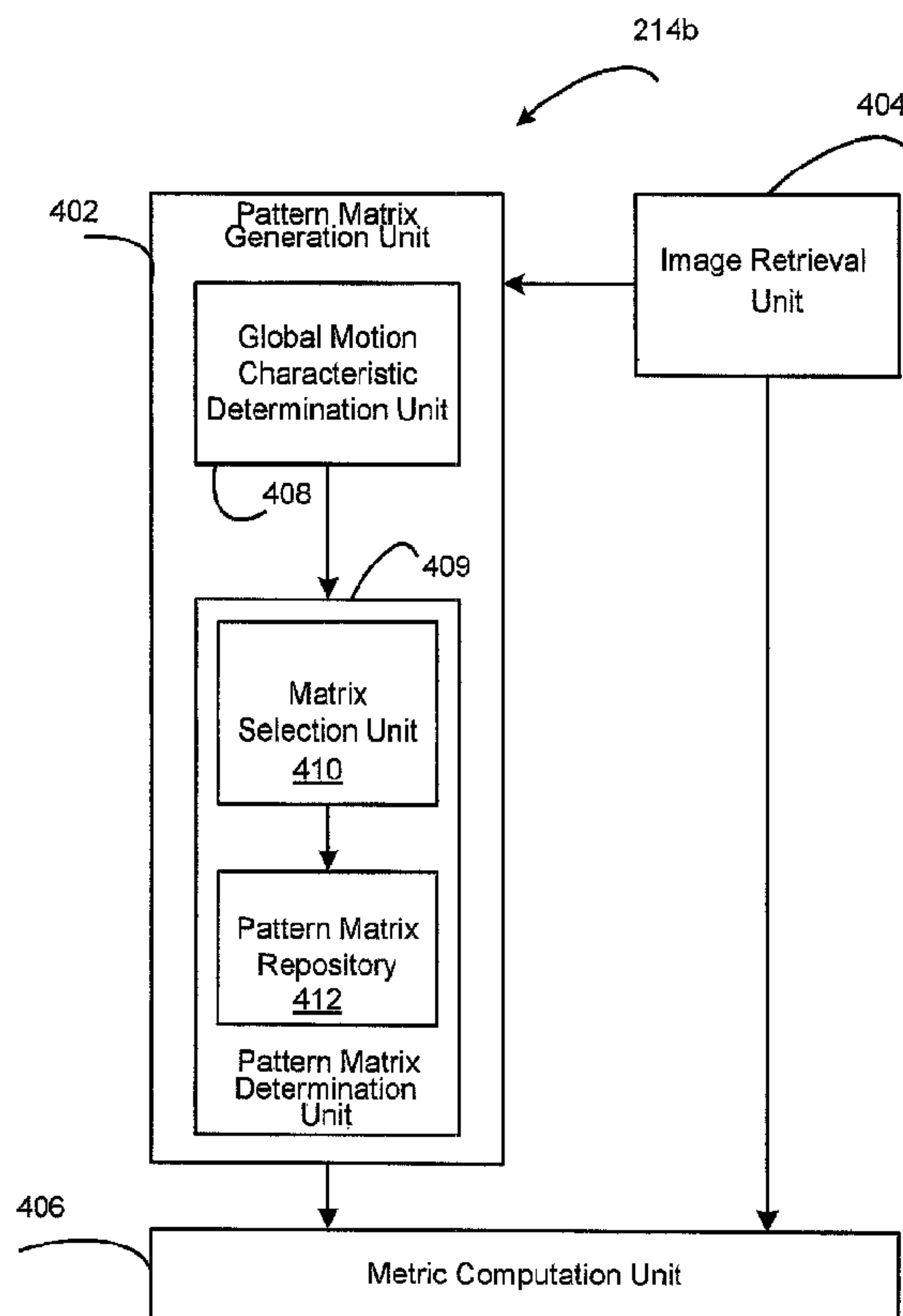




(86) Date de dépôt PCT/PCT Filing Date: 2008/10/16
 (87) Date publication PCT/PCT Publication Date: 2010/02/11
 (45) Date de délivrance/Issue Date: 2014/09/30
 (85) Entrée phase nationale/National Entry: 2010/11/09
 (86) N° demande PCT/PCT Application No.: CA 2008/001821
 (87) N° publication PCT/PCT Publication No.: 2010/015066
 (30) Priorité/Priority: 2008/08/02 (US12/185,077)

(51) Cl.Int./Int.Cl. *H04N 19/137* (2014.01),
G06T 9/00 (2006.01), *H04N 19/182* (2014.01),
H04N 19/186 (2014.01), *H04N 19/527* (2014.01)
 (72) Inventeur/Inventor:
 PIGEON, STEVEN, CA
 (73) Propriétaire/Owner:
 ECOLE DE TECHNOLOGIE SUPERIEURE, CA
 (74) Agent: DONNELLY, VICTORIA

(54) Titre : PROCEDURE ET SYSTEME DE DETERMINATION D'UNE METRIQUE POUR COMPARER DES BLOCS D'IMAGE DANS UN CODAGE VIDEO COMPENSE EN MOUVEMENT
 (54) Title: METHOD AND SYSTEM FOR DETERMINING A METRIC FOR COMPARING IMAGE BLOCKS IN MOTION COMPENSATED VIDEO CODING



(57) Abrégé/Abstract:

Method and system for determination of a metric measuring a difference between two image blocks used in motion compensated video coding of scenes are described. Only selected pixels in a block in the scene are processed for enhancing the speed of the metric computation.



(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
11 February 2010 (11.02.2010)(10) International Publication Number
WO 2010/015066 A1(51) International Patent Classification:
H04N 7/50 (2006.01) *G06T 9/00* (2006.01)

(72) Inventor; and

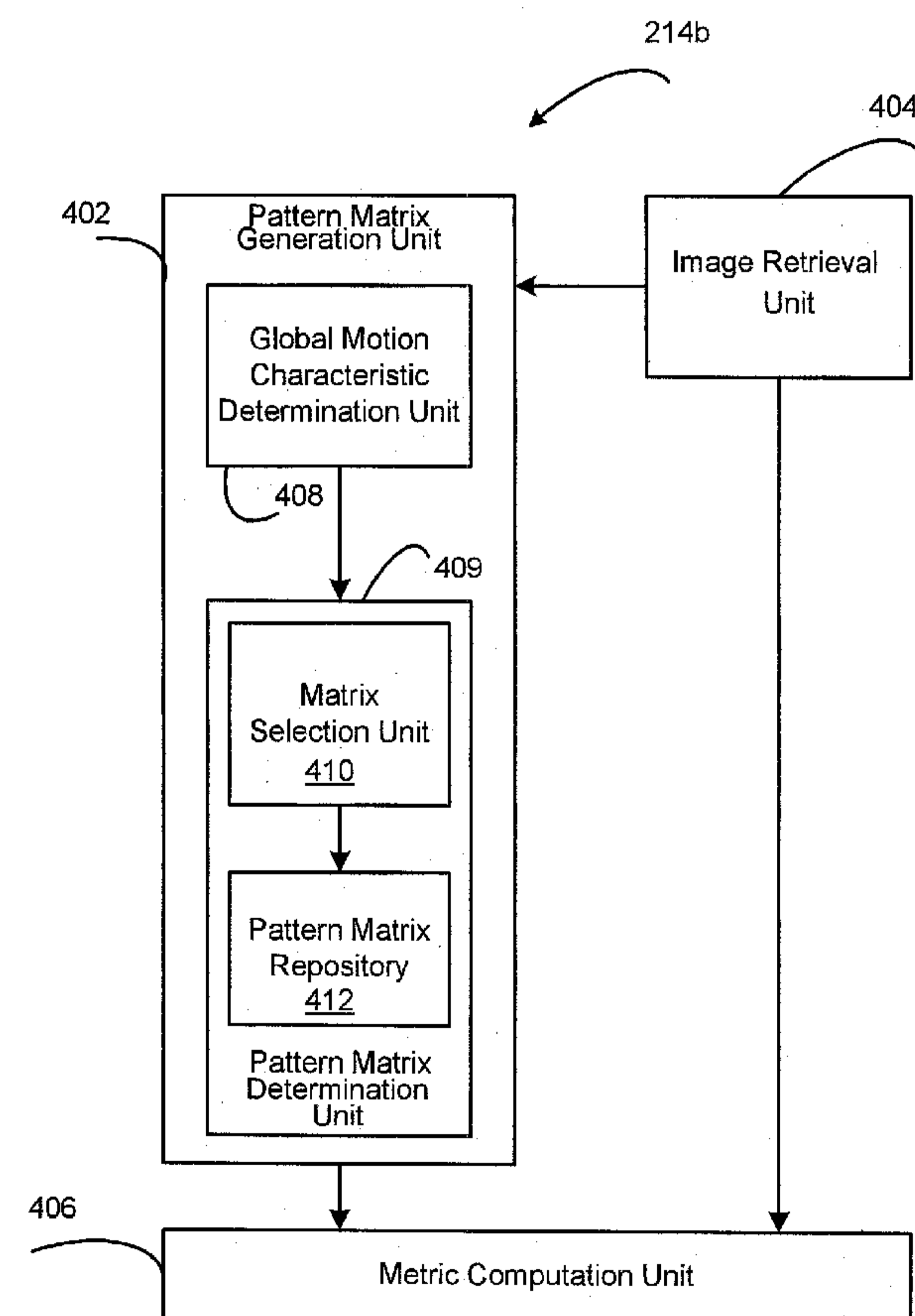
(75) Inventor/Applicant (for US only): **PIGEON, Steven**
[CA/CA]; 20, Yvan, Blainville, Quebec J7C 1Z9 (CA).(21) International Application Number:
PCT/CA2008/001821(74) Agent: **DONNELLY, Victoria**; PO Box 24001,
Hazeldean RPO, Kanata, Ontario K2M 2C3 (CA).(22) International Filing Date:
16 October 2008 (16.10.2008)(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ,
CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ,
EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,
HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR,
KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME,
MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO,
NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG,
SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA,
UG, US, UZ, VC, VN, ZA, ZM, ZW.

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/185,077 2 August 2008 (02.08.2008) US(71) Applicant (for all designated States except US): **ECOLE
DE TECHNOLOGIE SUPERIEURE** [CA/CA]; 1100
Notre-Dame Street Ouest, Montréal, Québec H3C 1K3
(CA).

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR DETERMINING A METRIC FOR COMPARING IMAGE BLOCKS IN MOTION
COMPENSATED VIDEO CODING(57) Abstract: Method and system for determination of a metric
measuring a difference between two image blocks used in motion
compensated video coding of scenes are described. Only selected
pixels in a block in the scene are processed for enhancing the
speed of the metric computation.**Figure 4 (c)**

WO 2010/015066 A1

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *with international search report (Art. 21(3))*

METHOD AND SYSTEM FOR DETERMINING A METRIC FOR COMPARING IMAGE BLOCKS IN MOTION COMPENSATED VIDEO CODING

FIELD OF THE INVENTION

The invention relates to the field of motion compensated video coding, and in particular, to an improved method and system for the determination of a metric used for comparing image blocks in motion compensated video coding.

BACKGROUND OF THE INVENTION

Multimedia containing various content types including text, audio and video, provides an outstanding business and revenue opportunity for network operators. The availability of higher bandwidth and the use of packet-switched Internet Protocol (IP) technology have made it possible to transmit richer content that include various combinations of text, voice, still and animated graphics, photos, video clips, and music. In order to capitalize on this market potential network operators must meet customers' expectations regarding quality and reliability. Transcoding of media at server level is crucial for rendering multimedia applications in today's heterogeneous networks composed of mobile terminals, cell phones, computers and other electronic devices. The adaptation and transcoding of media must be performed at the service provider level because individual devices are often resource constrained and are rarely capable of adapting the media themselves. This is an important problem for service providers, as they will have to face a very steep traffic growth in the next few years; growth that far exceeds the speed up one can obtain from new hardware alone. Using a brute-force approach of increasing the number of servers is not sufficient. Moreover, an increase in the number of servers leads to proportional increases in power consumption, heat dissipation and space. Another way to improve system performance and handle the large growth in traffic is to devise smart techniques for video coding that forms an important and resource intensive phase of multimedia adaptation.

Motion compensated video coding processes scenes consisting of blocks and each block consists of a number of pixels. Essentially all modern video codecs use motion compensated coding where frames are encoded relative to a number of preceding frames to exploit temporal dependencies and get better compression. The most intensive phase of movement compensated video coding is the movement estimation

phase. This is performed through a movement estimation algorithm that estimates the scene's objects displacements from one frame to the next. These estimations are used to create a synthetic frame where the scene is deformed to match the estimated movement of objects. That synthetic frame is used as a predictor for the current frame, which is differentially encoded. Such movement estimation algorithms are computationally intensive and account for a very large part of the encoder's runtime, increasingly so with resolution, making it a natural target for optimization.

A considerable amount of effort has been directed towards the problem of block-based movement estimation, a simplification to the general problem where the prediction frame is constructed from small rectangular regions copied from reference frames. A discussion of block-based movement estimation is provided next. For the explanation provided in this document we assume that the basic blocks are 16×16 pixels. Note that the same concepts are applicable for blocks of different sizes. The objective of the system is to produce a predicted frame for the current frame being encoded. This predicted frame is generated by differentially encoding the current frame from a given reference frame. For each 16×16 block in the current frame, the system looks for the best matching block in the reference frame. The search examines a number of blocks (not necessarily aligned on 16×16 boundaries) in the reference frame and selects the block that minimizes the difference with the current block. The motion vector, a key element in the motion estimation process, is simply the offset to the best matching block (in the reference frame) relative to the current block's position (in the current frame). The best matching block is then copied into the compensated frame or predicted frame at the current block's position. After this process, the predicted frame is the best approximation (according to the chosen metric measuring the difference between image blocks) one can build from the reference frame considering that only block copies are allowed. The compensated frame is used as the predictor to differentially encode the current frame.

A brief discussion of selected prior art references is presented below. Research has taken a number of different directions. S. Borman, M. Robertson, R.L Stevenson "Block Matching Sub-Pixel Motion Estimation from Noisy, Undersampled Frames" SPIE Visual Communications and Image Processing Conference 1999, presents an empirical study that concerns the effects of noise or sampling error in SAD, MSE, and NCF. The

paper, W. Li, E. Salari, "Successive Elimination Algorithm for Motion Estimation", IEEE Transactions on Image Processing, Volume 4, Issue 1, January 1995, pages 105-107, explores the properties of SAD and MSE for devising a dynamic-programming like method for fast motion estimation. The authors focus on an algorithm, which does not require an exhaustive search in the solution space and discusses how properties of existing metrics are to be used; they do not propose any new metric. F. Tombari, S. Mattocia, L. di Stefano, "Template Matching Based on Lp Norm Using Sufficient Conditions with Incremental Approximation", IEEE International Conference on Video and Signal Based Surveillance, November 2006, page 20, extends the work of Li and Salari. The paper uses a similar dynamic-programming approach to compute a fast version of a metric.

U. Koc and K.J.R. Liu, "Interpolation-free Subpixel Motion Estimation Technique in DCT Domain", IEEE Transactions on Circuits and Systems for Video Technology, Volume 8, Issue 4, August 1998, pages 460-487 focuses on a subpixel level and tries to avoid subpixel interpolation in the space domain by using techniques in the DCT domain that are at least as complex as the techniques used in the space domain. The metric is extended appropriately for handling the shift to the DCT domain. Another paper of S. Lee, Soo-ik Chae, "Two-step Motion Estimation Algorithm using Low Resolution Quantization", International Conference on Image Processing, Volume 3, September 1996, pages 795-798, focuses on motion estimation techniques. This paper presents a "fail fast" approach to SAD matching. The image is first quantized so that the precision of each pixel is reduced, for example from 8 bits per pixels to 4 bits per pixel. A first function compares the two blocks using the reduced precision version. If the results are acceptable, it proceeds to using a full precision metric. Although the research is presented with a hardware implementation in mind, it does not consider the effective utilization of a Single Instruction Multiple Data (SIMD) instruction set that includes SAD when the processor running the code provides such a facility. An important aspect of this invention is to reduce the time required in the computation of the metric by using such performance optimizing SIMD instruction sets that are provided in commercial processors available in the market today.

The research reported in C.-K. Cheung, L.-M. Po, "A Hierarchical Block Motion Estimation Algorithm using Partial Distortion Measure" International Conference on

Image Processing, Volume 3, October 1997, pages 606 - 609 uses pixel sampling by using regular grid sampling, which is strictly equivalent to ordinary sub-sampling. They compute SAD/MSE using $1/2$ or $1/4$, of the pixels (either in a quincunx pattern, or one in two columns, one in two rows). Blocks are checked against a $1/4$ grid SAD. If it is among the n better ones, it is kept for the next round, when a $1/2$ grid density will be used. Of the n better ones obtained from the previous round, m will be retained, and thoroughly checked with a full SAD. Unfortunately, the approach proposed by Cheung and Po cannot effectively utilize SIMD type parallel operations.

The research reported in Y.-L. Chan, W.-C. Siu, "New Adaptive Pixel Decimation for Block Motion Vector Estimation", IEEE Transactions on Circuits and Systems for Video Technology, Volume 6, Issue 1, February 1996, pages 113-118 is similar to the paper by Cheung and Po. However, Chan and Siu use different sampling patterns: regular, excluding quincunx. They consider patterns of density $1/4$ and $1/9$ (1 in 2×2 or one in 3×3), and they are not concerned with sub-pixel estimation.

Thus, various types of the metric measuring the difference between image blocks, to be referred to as the metric in the following discussion, have been used in existing codecs for block comparison. Irrespective of the exact metric used, its computation turns out to be computationally expensive.

Therefore, there is a need in the industry for an improved and effective method and system for fast computation of the metric measuring the difference between image blocks.

SUMMARY OF THE INVENTION

Therefore there is an object of the present invention to provide an improved method and system for the computation of the metric measuring the difference between two image blocks used for comparing blocks during motion compensated video coding.

According to one aspect of the invention, a method for determining a metric used in motion compensated video coding of a scene comprising blocks having pixels, the metric measuring a difference between a first image block and a second image block, the method

comprising: generating a pattern matrix, having the same number of rows and columns as the first and second image blocks, for specifying a subset of pixels to be selected in the first and second image blocks; dividing the subset of pixels into one or more groups of successive pixels, each group of successive pixels comprising an uninterrupted sequence of pixels in a row; a number of pixels in each group of successive pixels being selected so that to be operable in a single instruction of a Single Instruction Multiple Data (SIMD) processor; obtaining characteristics of pixels in the subset of pixels for the first and second image blocks; and processing the obtained characteristics for determining the metric.

The step of generating further comprises generating the pattern matrix based on a pixel budget, which is an estimate of number of pixels in the subset of pixels.

The step of generating further comprises selecting the pattern matrix from a repository of pattern matrices generated prior to performing motion compensation.

The step of selecting further comprises: for each pattern matrix in the repository of pattern matrices: storing boolean values in cells of the pattern matrix, each cell corresponding to pixels occupying the same position in the first and second image blocks; and storing one of the "true" or "false" boolean values in those cells of the pattern matrix, which correspond to the pixels from the first and second image blocks to be included in a computation of the metric, and storing the other boolean value in those cells of the pattern matrix, which correspond to the pixels in the first and second image blocks to be excluded from the computation of the metric.

In the described method, each pattern matrix corresponds to the SIMD processor architecture used for the motion compensated video coding, and a global motion characteristic of the scene indicating a direction of movement of an object in the scene.

In the described method, the step of obtaining comprises obtaining a luminance value of each pixel.

The method further comprises generating a global motion characteristic of the scene indicating a direction of movement of an object in the scene; and generating the pattern matrix corresponding to an architecture of the SIMD processor used for the motion compensated video coding and the global motion characteristic.

The step of generating further comprises generating the pattern matrix based on a pixel budget, which is an estimate of number of pixels in the subset of pixels.

In the described method, the step of generating the global motion characteristic further comprises: determining an average motion vector for the scene, the average motion vector having an amplitude and a direction; comparing the amplitude of the average motion vector with a threshold of no movement, comprising: determining the global motion characteristic as static signifying no object movement when the amplitude of the average motion vector is below the threshold; and determining the global motion characteristic as one of horizontal movement, vertical movement, or diagonal movement based on the direction of the average motion vector when the amplitude of the average motion vector is equal to or above the threshold.

In the described method, the step of generating further comprises selecting the pattern matrix from a repository of pattern matrices generated prior to performing motion compensation.

In the described method, the step of selecting further comprises: for each pattern matrix in the repository of pattern matrices: storing boolean values in cells of the pattern matrix, each cell corresponding to pixels occupying the same position in the first and second image blocks; and storing one of the "true" or "false" Boolean values in those cells of the pattern matrix, which correspond to the pixels from the first and second image blocks to be included in a computation of the metric, and storing the other boolean value in those cells of the pattern matrix, which correspond to the pixels in the first and second image blocks to be excluded from the computation of the metric.

In the method described above, the step of obtaining further comprises: obtaining the characteristic, which is a luminance value; storing luminance values of pixels in the first image block in cells in a first matrix; and storing luminance values of pixels in the second image block in cells in a second matrix.

In the method described above, the step of processing further comprises: selecting cells in the first and the second pattern matrices using the pattern matrix; and applying one of a Sum of Absolute Differences function or a Mean Squared Error function on the luminance values stored in the selected cells.

In the method described above, positions of said one or more groups of successive pixels are selected to as to maximize a span of block coverage.

According to another aspect of the invention, there is provided a system for determining a metric used in motion compensated video coding of a scene comprising blocks having pixels, the metric measuring a difference between a first image block and a second image block, the system comprising: a pattern matrix generation unit, producing a pattern matrix, having the same number of rows and columns as the first and second image blocks, for specifying a subset of pixels to be selected in the first and second image blocks, the subset of pixels being divided into one or more groups of successive pixels, each group of successive pixels comprising an uninterrupted sequence of pixels in a row; a number of pixels in each group of successive pixels being selected so that to be operable in a single instruction of a Single Instruction Multiple Data (SIMD) processor; an image retrieval unit, retrieving characteristics of pixels in the subset of pixels for the first and second image blocks; and a metric computation unit, determining the metric by processing the retrieved characteristics.

In the system described above, the pattern matrix generation unit is configured to generate the pattern matrix based on a pixel budget, which is an estimate of number of pixels in the subset of pixels.

In the system described above, the pattern matrix generation unit comprises a repository of pattern matrices generated prior to performing motion compensation.

In the system described above, each pattern matrix in the repository of pattern matrices is adapted to: store boolean values in cells of the pattern matrix, each cell corresponding to pixels occupying the same position in the first and second image blocks; and store one of the “true” or “false” Boolean values in those cells of the pattern matrix, which correspond to the pixels from the first and second image blocks to be included in a computation of the metric, and store the other boolean value in those cells of the pattern matrix, which correspond to the pixels in the first and second image blocks to be excluded from the computation of the metric.

In the system described above, each pattern matrix corresponds to an architecture of the SIMD processor used for the motion compensated video coding, and a global motion characteristic of the scene indicating a direction of movement of an object in the scene.

In the system described above, the characteristic is a luminance value.

In the system described above, the pattern matrix generation unit further comprises: a global motion characteristic determination unit, computing a global motion characteristic of the scene indicating a direction of movement of an object in the scene; and a matrix determination unit, generating the pattern matrix based on the global motion characteristic of the scene, an architecture of the SIMD processor used in the motion compensated video coding.

In the system described above, the matrix determination unit is further adapted to generate the pattern matrix based on a pixel budget, which is an estimate of number of pixels in the subset of pixels.

In the system described above, the matrix determination unit further comprises: a pattern matrix repository, storing a series of predetermined pattern matrices, each pattern matrix in the pattern matrix repository corresponding to the SIMD processor architecture used in the

motion compensated video coding, the global motion characteristic of the scene, and the pixel budget; and a pattern matrix selection unit, selecting the pattern matrix from the pattern matrix repository.

In the system described above, the metric computation unit comprises one of a Sum of Absolute Differences computing unit, applying the Sum of Absolute Differences function on values of the characteristic in the subset of pixels, or a Mean Squared Error computing unit, applying the Mean Squared Error function on values of the characteristic in the subset of pixels.

In the system described above, positions of said one or more groups of successive pixels are selected so that as to maximize a span of block coverage.

Thus, an improved system and method for determining a metric used in motion-compensated video coding have been provided.

BRIEF DESCRIPTION OF THE DRAWINGS

Further features and advantages of the invention will be apparent from the following description of the embodiment, which is described by way of example only and with reference to the accompanying drawings, in which:

Figure 1 illustrates a prior art system for motion compensated video coding;

Figure 2 presents the Core Subsystem 102a of Figure 1 in more detail;

Figure 3 shows a flowchart illustrating steps of the prior art method for motion estimation in compensated video coding;

Figure 4(a) presents an example for illustrating successive pixels selection based on the CPU architecture for reducing the computation cost of metric determination according to the embodiment of the present invention;

Figure 4(b) presents an Improved Core Subsystem 102b of the embodiment of the present invention;

Figure 4(c) shows units in the Improved Metric Determination Module 214b of Figure 4(b);

Figure 5 shows a flowchart illustrating the steps of the method executed by the embodiment of the invention for determining the metric measuring the difference between image blocks;

Figure 6 illustrates the procedure used for determining the global motion characteristic of the scene in the embodiment of the invention; and

Figure 7 shows a flowchart illustrating the step "Compute P-SAD" of Figure 5.

DETAILED DESCRIPTION OF THE EMBODIMENT OF THE INVENTION

Because block-based motion estimation algorithms are based on the approximately correct assumption that the metric measuring the difference between two image blocks (often referred to in the literature as error) is concave around the optimal region, they depend only loosely on the actual metric and therefore should be resilient to efficient, low-cost, approximated metrics. This invention reduces the computational cost associated with the determination of the metric measuring the difference between image blocks by considering only selected pixels in the blocks being compared. Although the time required for determining the metric is reduced drastically, there is no significant degradation in the quality of the image.

Before discussing the invention in detail, a general description of motion compensated encoding is provided with the help of the system 100 shown in Figure 1. The Core subsystem 102a includes a Frame Buffer module 104 that is connected to the Motions Estimation module 106 and the Motion Compensation module 108. The Motion Estimation module 106 is connected in turn to the Motion Compensation module 108. The Frame Buffer module 104 stores a number of previous frames that are required by the other modules to create motion-compensated frames that are used as predictors for the current frame under processing. Further details of the Core subsystem 102a that produces a motion compensated image are described in detail in Figure 2.

The motion-compensated frame from the Motion Compensation module 108 is subtracted from the current image (shown by the minus operator 114 in Figure 1) and the residual (the image difference) is sent to the transform step for encoding.

Frames are produced by a Frame Source 110 that can be any apparatus (e.g., a camera or a file) that feeds image frames into the encoder. The processing of the frames stored in Frame Source 110 depends on the mode selected. The Mode Selection module 112 indicates to the codec whether the incoming frame is to be coded with motion compensation or an "intra mode" is to be used. An intra mode is used with frames (e.g., as key frames) that cannot be decoded relative to other frames. They use some standalone coding that is limited to the data entirely contained within that image. Mode selection is external to motion compensation as this is a stream-level policy-based feature. Mode selection selects between interpolated frames (motion-compensated) or "key-frames" that can be decoded on their own, without compensation

or reference to other frames. Key-frames are used, among other things, to seek to a specific position in the video without decoding previous frames. Decompression can only start at key-frames, while motion-compensated frames depend on other frames and therefore cannot be the initial decompression point. When the Mode Selection module 112 selects the Intra Prediction module 116, this module 116 predicts the pixels within the current frame without the help of other frames. This prepares a key-frame for encoding. As mentioned earlier, the intra-coded frame does not depend on other frames for its compression/decompression. Further details of intra prediction are beyond the scope of the present discussion. If the Mode Selection module 112 selects to perform the transform operation, the frames after being processed in accordance with the output of the Motion Compensation module 108 are fed into the Transform module 118. Transformation is a codec-specific step, where the image is transformed from the spatial (pixel) domain to a frequency domain. This transform is usually the discrete cosine transform (DCT) or a kin transform. Transform coding is well known, and further discussion is not needed in the context of the present invention. After transforming the pixels into a frequency domain, the resulting data is quantized with the help of a Quantization Module 128. Quantization is basically a precision-reduction (therefore irreversible) step, which, in practice, means that fewer bits will be used to represent the data. The type and coarseness of quantization depend on the codec and the user-specified target quality/bit rate, respectively. The output of the Quantization module 128 is processed by the Inverse Quantization Module 126 and the Entropy Coding module 130 that also receives the motion vectors from the Motion Estimation module 106. Entropy coding is a codec-specific step where reduced precision data from the Quantization module 128 are encoded by using a variable length code and other compression techniques. At this level, no precision is lost, only a more efficient representation of data is used. Typically, it is a variation on Huffman coding, and uses either a static code that is precomputed, or some adaptive code that evolves as it codes the data. For more advanced codecs, such as H.264, more sophisticated techniques are used.

The output of the Entropy Coding module 130 is processed by the Transport Coding module 132. This module "wraps" the entropy-coded data into a transport format based on whether the output is stored as a file in a Storage module 136 or transmitted as data by the Transmit module 134. Typical transport file formats are MPEG TS, AVI, 3GP,

etc. Transmission could be over RTP/RTSP, a real time transport protocol used to stream video over the Internet.

The Inverse Quantization module 126 receives the output of the Quantization module 128 and undoes the step performed by the Quantization module 128. However, this is not a perfect inverse, as quantization irremediably removes information; the Inverse quantization produces a reduced-precision approximation of the original data. This data is then processed by the Inverse Transform module 124. This module inverses the transform step and converts data from frequency domain to the pixel domain. At this point, one has a reduced precision approximation of the image that went in the Transform module 118.

At the upper center of the Figure 1, there is a second Mode Selection switch 120 (inverse of the switch 112) that decides whether the image to add to the Frame Buffer module 104 is an intra frame or a motion-compensated frame. If it is an intra frame, then the output of the Inverse Transform module 124 is already a complete image and can therefore be added as is in the Frame Buffer module 104. If it is a motion compensated frame, the difference from the inverse transform is added back to the motion-compensated frame (the operation is symbolized by the plus operator 122 in the figure), resulting in the final coded image that is added to the Frame Buffer module 104.

The Core subsystem 102a is explained with the help of Figure 2. The Last Encoded Image 202 is an image produced by the Inverse Transform module 124 of Figure 1 and is inserted in the Frame Buffer module 104 that stores a number of frames. Typically, the Frame Buffer module 104 is limited to storing very few Buffered Frames 208, that typically include the last one or two encoded images. When a new frame is added, the oldest frame (the one that has been in the buffer for the longest time) is discarded, ensuring that the buffer does not grow indefinitely. In certain codecs, such as H.264, however, the number of frames can be as high as 16, but in general it is limited to the previous frame only. The Select Frame module 210 selects the appropriate frame for the next module depending on Frame Selection output from the Motion Estimation Algorithm 216.

The Current Image to Encode 212 and the frame selected by the Select Frame module 210 are processed by the Motion Estimation module 106 that deploys the Motion Estimation Algorithm 216 that performs the actual motion estimation. It uses the Metric Determination module 214a, producing the metric used to measure the difference between two image blocks. Depending on the complexity of the Motion Estimation Algorithm 216, frame selection can be as simple as "obtain the last frame" or more complex as in the selection of half-frames in an interlaced mode. Frame selection also depends on the specific codecs supported. Some will support only the last frame as a reference frame, but some codecs, such as H.264, allow patches to come from a large number of images (up to 16 in this case). This module returns the motion vectors (which may also include references to multiple previous frames).

The output of the Motion Estimation Algorithm 216 is used by the Motion Compensation module 108. This module applies the transformations described by the motion vectors (and references to previous frames). The Motion Compensation module 108 stitches a prediction image from parts of the reference frame(s) as specified by the motion vectors.

The steps of the method executed by a typical motion estimation algorithm is explained with the help of flowchart 300 presented in Figure 3. Upon start (box 302), the procedure 300 sets the variable best metric to an arbitrary large value (box 304) and initializes the search position (box 306). Search position is the position of the block in the reference frame that is compared to the current block to compensate in the current frame that is being processed. The motion estimation algorithm generates a number of such positions in the reference frame and tests blocks at those positions by comparing them to the current block in the current frame. The procedure 300 then tests the search position (box 308). The search position is tested by applying the Sum of Absolute Differences (SAD) function on the luminance value of each pixel stored in the corresponding cells of the two matrices a and b. The block from the reference frame being tested is referred to as the first block and the luminance values of its pixels are stored in a first matrix referred to by symbol a in the following equations. The block from the current frame that is being processed is referred to as the second block and the luminance values of its pixels are stored in a second matrix referred to by symbol b in

the following equations. A cell in row i and column j of the two matrices are referred to as $a[i,j]$ and $b[i,j]$ respectively. Thus,

$$\text{SAD} = \sum_{(i,j)} | a[i,j] - b[i,j] |$$

The magnitude of the resulting difference between the two blocks as computed by this SAD function, and referred to as the metric, is compared with best metric (box 310). If metric is lower than best metric then the procedure 300 exits "Yes" from box 310 and computes the motion vector (box 314) from this search position. Otherwise, the procedure 300 exits "No" from box 310 and checks whether or not additional search positions need to be generated (box 312). If not, the procedure 300 exits "No" from box 312 and exits (box 322). Otherwise, a new search position is generated (box 315) and the procedure 300 loops back to the entry of box 308. After computing the motion vector in box 314, the procedure 300 stores the value of metric in best metric and the motion vector computed in Box 314 in a variable called best vector (box 316). Whether best metric is lower than a predetermined threshold is checked next (box 318). If not, the procedure 300 exits "No" from box 318 and loops back to the entry of box 312. Otherwise, the procedure 300 exits "Yes" from box 318, adds best vector to the motion vector buffer (box 320) and exits (box 322).

Computation of the metric measuring the difference between two image blocks by using a SAD function, for example, is time consuming because it is based on every pixel in the two blocks. The current invention reduces the computational complexity significantly by deploying two different techniques. Only a selected subset of pixels are used in the computation of the metric:

$$\text{P-SAD} = \sum_{(i,j \in P)} | a[i,j] - b[i,j] |$$

P-SAD is the optimized pattern based SAD function that uses only the subset of pixels that are included in P and is based on a pattern matrix used for the selection of the subset of pixels. The computational complexity of P-SAD is directly proportional to the number of i,j pairs that are included in the computation. The pixels (identified by i,j) are selected in such a way that the distortion in the resulting image is negligibly small. The P-SAD procedure accepts three arguments: the a matrix, the b matrix and a pattern

matrix. Pattern is a boolean matrix and includes a "true" or "false" at a particular combination of i and j to indicate whether or not a particular set of pixel luminance values, $a[i,j]$ and $b[i,j]$, is to be included in the computation of P-SAD: a value of "true" for $pattern[i,j]$ indicates that the corresponding $a[i,j]$ and $b[i,j]$ pair is included in the computation of P-SAD whereas a value of "false" means that the corresponding $a[i,j]$ and $b[i,j]$ pair is excluded from this computation. By reducing the number of terms that are used in the computation of P-SAD, the computational complexity is greatly reduced. The execution time for the computation of P-SAD is further optimized by selecting successive pixels in a row referred to as a group. A number of available CPUs can operate on such a group in a single CPU instruction. Several CPU's existing in the market include a Single Instruction, Multiple Data (SIMD) instruction set for performing such an operation. Examples include Intel's Pentium III series processors that include the Streaming SIMD Extensions (SSE) to the x86 architecture. 3DNow! is a multimedia extension for AMD's K6 processor that provides the packed SAD from byte summed to word (PSADBW) instruction.

An example with five groups is shown in diagram 350 displayed in Figure 4(a) where the gray cells indicate the selected pixels that are used in the computation of P-SAD for an example 16 x 16 block. Each group in this example includes eight pixels. Please note that a pattern matrix is constructed in such a way that the groups are as equidistant as possible (in both the horizontal and vertical directions) and the span of block coverage is maximized (see Figure 4(a)). The vertical distances between any two adjacent groups of Figure 4(a) are approximately equal whereas the horizontal distances between any two adjacent groups are equal to one another. With such positions of groups within a block, the span of block coverage is maximized. That is, the groups of Figure 4(a) are not clustered in one region of the block, but are placed in a way to cover as much of the block as possible. An Intel Pentium III processor with its SSE, for example, can operate on a group represented by a set of eight successive gray cells occupying the same row in one CPU instruction.

The number of "true" values (or alternatively, "false" values) in the pattern matrix directly controls the cost associated with the computation of P-SAD, and is based on a pixel budget. The pixel budget is an estimate of the number of pixels in the selected subset of pixels processed by P-SAD. It may have a predetermined fixed value or may be

specified by the user. The pattern matrix is chosen based on this pixel budget, the CPU architecture and the global motion characteristic of the scene being processed. The example of Figure 4(a) corresponds to a pixel budget of 40. Assuming that an Intel Pentium III processor with its SSE is used to operate on a group comprising eight pixels in one instruction, the number of groups is $40/8 = 5$ and thus five groups are used in the example of Figure 4(a).

The embodiment of the current invention reduces the computational complexity of metric determination by improving the Metric Determination module 214a. An Improved Core Subsystem 102b that incorporates an Improved Metric Determination module 214b implementing the P-SAD function is shown in system 370 of Figure 4(b) of the embodiment of the invention. Note that all components of the Improved Core Subsystem other than 214b are the same as those shown in the Core Subsystem 102a of Figure 2. The system 370 of embodiment of invention includes a general purpose or specialized computer having a CPU and a computer readable medium, e.g., memory, DVD, CD-ROM or other medium, storing computer readable instructions for execution by the CPU. Alternatively, the system can be implemented in firmware, or combination of firmware and a specialized computer. Various modules of the system 370, including the Improved Metric Determination Module 214b, are implemented as computer readable instructions stored in the computer readable medium for execution by the CPU.

The Improved Metric Determination module 214b is explained with the help of diagram 214b presented in Figure 4(c). As mentioned above, the Improved Metric Determination module 214b can be implemented in firmware, or alternatively, a computer software code stored in a computer readable medium.

The Improved Metric Determination Module 214b includes three units: the image retrieval unit 404, the pattern matrix generation unit 402 and the metric computation unit 406, comprising computer readable instructions stored in a computer readable medium. The characteristics of all pixels in the scene, such as their luminance values, are retrieved by the image retrieval unit 404 for performing motion compensated video coding. These characteristics are used by the pattern matrix generation unit 402 to generate an appropriate pattern matrix and by the metric computation unit 406 that

determines the metric measuring a difference between a first image block contained in the reference frame and a second image block contained in the current frame that is being processed. In one embodiment of the invention, the metric computation unit comprises a Sum of Absolute Differences computation unit implementing function P-SAD discussed earlier. The pattern matrix generation unit 402 in turn includes two units: a global motion characteristic determination unit 408 and a pattern matrix determination unit 409, comprising computer readable instructions stored in a computer readable medium. The global motion characteristic determination unit 408 computes a global motion characteristic of the scene that indicates the direction of movement of an object in the scene. This information is used by the matrix determination unit 409 for selecting the pattern matrix to be used in motion compensated video coding of the scene. The pattern matrix determination unit 409 includes two units: matrix selection unit 410 and a pattern matrix repository 412 stored in a computer readable medium. The pattern matrix repository 412 stores a series of predetermined pattern matrices each of which corresponds to a CPU architecture and the global motion characteristic of the scene. The pattern matrix selection unit 410 is responsible for selecting the pattern matrix from the pattern matrix repository 412 based on the CPU architecture used in motion compensated video coding, the global motion characteristic of the scene being processed as well as the pixel budget. The pattern matrix repository includes a number of pattern matrices each of which corresponds to a specific combination of CPU architecture used in motion compensated video coding, the global motion characteristic of the scene and pixel budget. For the example presented in the context of Figure 4(a), the pattern matrix repository will include four specific pattern matrices that correspond to an Intel Pentium III processor with its SSE extension and a pixel budget of 40. Each of these four pattern matrices corresponds to a specific global motion characteristic of the scene: static, horizontal movement, vertical movement and diagonal movement. Once the values for all the three factors are known the specific pattern matrix can be located and retrieved from the repository. The methods used for achieving these units are discussed next.

The steps of the method to compute the value of P-SAD performed by the metric computation unit 406 are explained with the help of flowchart 500 presented in Figure 5. Upon start (box 502), In_group , the number of pixels in a group, is chosen based on the CPU architecture (box 504). Recall that a group refers to successive pixels in a row

CA 2723910

REPLACEMENT SHEET

VT-046-CA

that are processed by one CPU instruction. For example, if the CPU supports instructions that can process n pixels at a time, ln_group is set to n . The number of such groups to be used referred to as no_group , is then computed (box 506). The Floor function produces the largest integer that is equal to or less than its argument. Based on ln_group , no_group and global motion characteristic of the scene, the pattern matrix to be used is determined (box 508). In one embodiment of the invention the pattern matrix is chosen from a set of stored matrices. Once the pattern matrix is determined, the procedure 500 computes the P-SAD value (box 510) and exits (box 512). The storage of various pattern matrices and its selection in box 508 are discussed next. This is followed by a discussion of the method for computing P-SAD.

Generating a pattern matrix is achieved by the pattern matrix generation unit 402. The selection of the pattern matrix is briefly discussed. A scene can be characterized in many different ways. One embodiment of the invention uses the global motion characteristic of a scene to characterize it as static, vertical movement, diagonal movement or horizontal movement. Note that this characterization captures the approximate direction of object movement, with static being a special case that corresponds to little or no movement. Diagram 600 displayed in Figure 6 is used to explain the procedure used in the determination of the global motion characteristic of a scene. The average motion vector, having an amplitude and a direction, for the scene is computed by adding the motion vectors for the different blocks in the scene and dividing the sum by the number of blocks. The amplitude (or length) of the average motion vector is compared to a given "threshold of no movement" 602 that is set by the user. If the vector amplitude is lower than this threshold value, the scene is classified as static. For example, the average motion vector 604 has an amplitude lower than the threshold of no movement and is determined static. Similarly because the amplitude of the average motion vector 606 is larger than the threshold of no movement, its direction is considered.

The scene characterization space is divided into eight regions in the example of Figure 6: regions 612 and 620 correspond to horizontal movement, regions 608 and 616 to vertical movement and regions 610, 614, 618, and 622 to diagonal movement. Depending on the region to which the average motion vector belongs, the scene is classified as horizontal movement, vertical movement or diagonal movement (see

Figure 6). In Figure 6, since the amplitude of the average motion vector 606 is larger than the threshold of no movement, and it lands in a vertical movement region 608 of the diagram, the scene is classified as vertical movement. In one embodiment of the invention a number of pattern matrices that are to be used in the computation of P-SAD are generated and stored in a repository prior to performing motion compensated video coding. Each such pattern matrix corresponds to a particular combination of *In_group*, *no_group* and the global motion characteristic of the scene. At run time, the average motion vector for the scene being processed is analyzed, and an appropriate pattern matrix is then retrieved from the repository of pattern matrices (box 508) and used in the computation of P-SAD that is described next.

The steps of the method executed in box 510 of Figure 5 are explained further with the help of flowchart 700 shown in Figure 7. The explanation is presented in the context of a system that uses image blocks having 16 x 16 pixels and can be easily extended to images with other blocks sizes. Upon start (box 702), the procedure 700 performs the initialization step (box 704). In this step, the initial values for *i* and *j* that store the indices of the current row and column respectively are set to 1 and the variable *sum* is initialized to 0. After the initialization step, the procedure 700 checks the boolean value stored in cell at the *i*th row and *j*th column of the pattern matrix (box 706). If it is "true", the procedure 700 exits "Yes" from box 706 and *a*[*i,j*], *b*[*i,j*] are used in the computation of the P-SAD as shown in box 708. Otherwise, the procedure 700 exits "No" from box 706 and the *a*[*i,j*], *b*[*i,j*] values are excluded from the computation of P-SAD. The operation performed in box 708 is thus skipped and the procedure 700 checks whether or not all the columns for the current row are considered (box 710). If not, the procedure 700 exits "No" from box 710, increments the value of *j* (box 712) to go to the next column and loops back to the entry of box 706. Otherwise, the procedure 700 exits "Yes" from box 710 and checks whether or not all the rows have been covered (box 716). If all rows have been covered, the procedure 700 exits "Yes" from box 716, returns the value in *sum* (box 718) and exits (box 720). Otherwise, the procedure 700 exits "No" from box 716, increments the value of *i* and sets the value of *j* to 1 (box 714) to go to the next row and loops back to the entry of box 706.

Although specific embodiments of the invention have been described in detail, it should be understood that the described embodiments are intended to be illustrative and not

restrictive. Various changes and modifications of the embodiments shown in the drawings and described in the specification may be made within the scope of the following claims without departing from the scope of the invention in its broader aspect. For example, instead of using the P-SAD function in determination of the metric, an optimized pattern based Mean Squares of Errors (P-MSE) function may be used:

$$\text{P-MSE} = \sum_{(i,j) \in P} (a[i,j] - b[i,j])^2$$

As in the case of P-SAD only selected pixels characteristics are used in the computation of the metric. Other functions can also be used in determining the metric, e.g., decibels, as required. Also, instead of the Floor function, a Ceiling function may be used in the determination of no_group (see Figure 5) when using a slightly higher number of pixels than specified by the pixel budget can be tolerated by the user. In certain embodiments, different units from those shown in Figure 4(c) may be used in the Improved Metric Determination module 214b for achieving the desired functionality. Moreover, such units may be executed on one computer or distributed over multiple computers.

A computer readable medium, storing instructions thereon for performing the steps of the methods of the embodiments of the invention, comprising a computer memory, DVD, CD-ROM, floppy or the like, is also provided.

Various other modifications may be provided as needed. It is therefore to be understood that within the scope of the given system characteristics, the invention may be practiced otherwise than as specifically described herein.

Although specific embodiments of the invention have been described in detail, it should be understood that the described embodiments are intended to be illustrative and not restrictive. Various changes and modifications of the embodiments shown in the drawings and described in the specification may be made within the scope of the following claims without departing from the scope of the invention in its broader aspect.

What is Claimed is:

1. A method for determining a metric used in motion compensated video coding of a scene comprising blocks having pixels, the metric measuring a difference between a first image block and a second image block, the method comprising:

generating a pattern matrix, having the same number of rows and columns as the first and second image blocks, for specifying a subset of pixels to be selected in the first and second image blocks;

dividing the subset of pixels into one or more groups of successive pixels, each group of successive pixels comprising an uninterrupted sequence of pixels in a row;

a number of pixels in each group of successive pixels being selected so that to be operable in a single instruction of a Single Instruction Multiple Data (SIMD) processor;

obtaining characteristics of pixels in the subset of pixels for the first and second image blocks; and

processing the obtained characteristics for determining the metric.

2. The method of claim 1, wherein the step of generating further comprises generating the pattern matrix based on a pixel budget, which is an estimate of number of pixels in the subset of pixels.

3. The method of claim 2, wherein the step of generating further comprises selecting the pattern matrix from a repository of pattern matrices generated prior to performing motion compensation.

4. The method of claim 3, wherein the step of selecting further comprises:

for each pattern matrix in the repository of pattern matrices:

storing boolean values in cells of the pattern matrix, each cell corresponding to pixels occupying the same position in the first and second image blocks; and

storing one of the "true" or "false" boolean values in those cells of the pattern matrix, which correspond to the pixels from the first and second image blocks to be included in a computation of the metric, and storing the other boolean value in those cells

of the pattern matrix, which correspond to the pixels in the first and second image blocks to be excluded from the computation of the metric.

5. The method of any one of claims 1 to 4, wherein each pattern matrix corresponds to the SIMD processor architecture used for the motion compensated video coding, and a global motion characteristic of the scene indicating a direction of movement of an object in the scene.

6. The method of any one of claims 1 to 4, wherein the step of obtaining comprises obtaining a luminance value of each pixel.

7. The method of claim 1, wherein the step of generating further comprises:

generating a global motion characteristic of the scene indicating a direction of movement of an object in the scene; and

generating the pattern matrix corresponding to an architecture of the SIMD processor used for the motion compensated video coding and the global motion characteristic.

8. The method of claim 7, wherein the step of generating further comprises generating the pattern matrix based on a pixel budget, which is an estimate of number of pixels in the subset of pixels.

9. The method of claim 7, wherein the step of generating the global motion characteristic further comprises:

determining an average motion vector for the scene, the average motion vector having an amplitude and a direction;

comparing the amplitude of the average motion vector with a threshold of no movement, comprising:

determining the global motion characteristic as static signifying no object movement when the amplitude of the average motion vector is below the threshold; and

determining the global motion characteristic as one of horizontal movement, vertical movement, or diagonal movement based on the direction of the average motion vector when the amplitude of the average motion vector is equal to or above the threshold.

10. The method of any one of claims 7 to 9, wherein the step of generating further comprises selecting the pattern matrix from a repository of pattern matrices generated prior to performing motion compensation.

11. The method of claim 10, wherein the step of selecting further comprises:
for each pattern matrix in the repository of pattern matrices:

storing boolean values in cells of the pattern matrix, each cell corresponding to pixels occupying the same position in the first and second image blocks; and

storing one of the "true" or "false" Boolean values in those cells of the pattern matrix, which correspond to the pixels from the first and second image blocks to be included in a computation of the metric, and storing the other boolean value in those cells of the pattern matrix, which correspond to the pixels in the first and second image blocks to be excluded from the computation of the metric.

12. The method of any one of claims 1, 2, 3, 4, 7, 8, 9, wherein the step of obtaining further comprises:

obtaining the characteristic, which is a luminance value;

storing luminance values of pixels in the first image block in cells in a first matrix;

and

storing luminance values of pixels in the second image block in cells in a second matrix.

13. The method of claim 12, wherein the step of processing further comprises:

selecting cells in the first and the second pattern matrices using the pattern matrix;

and

applying one of a Sum of Absolute Differences function or a Mean Squared Error function on the luminance values stored in the selected cells.

14. The method of any one of claims 1, 2, 3, 4, 7, 8, 9, wherein positions of said one or more groups of successive pixels are selected to as to maximize a span of block coverage.

15. A system for determining a metric used in motion compensated video coding of a scene comprising blocks having pixels, the metric measuring a difference between a first image block and a second image block, the system comprising:

- a pattern matrix generation unit, producing a pattern matrix, having the same number of rows and columns as the first and second image blocks, for specifying a subset of pixels to be selected in the first and second image blocks, the subset of pixels being divided into one or more groups of successive pixels, each group of successive pixels comprising an uninterrupted sequence of pixels in a row;

- a number of pixels in each group of successive pixels being selected so that to be operable in a single instruction of a Single Instruction Multiple Data (SIMD) processor;

- an image retrieval unit, retrieving characteristics of pixels in the subset of pixels for the first and second image blocks; and

- a metric computation unit, determining the metric by processing the retrieved characteristics.

16. The system of claim 15, wherein the pattern matrix generation unit is configured to generate the pattern matrix based on a pixel budget, which is an estimate of number of pixels in the subset of pixels.

17. The system of claim 16, wherein the pattern matrix generation unit comprises a repository of pattern matrices generated prior to performing motion compensation.

18. The system of claim 17, wherein each pattern matrix in the repository of pattern matrices is adapted to:

- store boolean values in cells of the pattern matrix, each cell corresponding to pixels occupying the same position in the first and second image blocks; and

store one of the “true” or “false” Boolean values in those cells of the pattern matrix, which correspond to the pixels from the first and second image blocks to be included in a computation of the metric, and store the other boolean value in those cells of the pattern matrix, which correspond to the pixels in the first and second image blocks to be excluded from the computation of the metric.

19. The system of any one of claims 15 to 18, wherein each pattern matrix corresponds to an architecture of the SIMD processor used for the motion compensated video coding, and a global motion characteristic of the scene indicating a direction of movement of an object in the scene.

20. The system of any one of claims 15 to 18, wherein the characteristic is a luminance value.

21. The system of claim 15, wherein the pattern matrix generation unit further comprises:
a global motion characteristic determination unit, computing a global motion characteristic of the scene indicating a direction of movement of an object in the scene;
and
a matrix determination unit, generating the pattern matrix based on the global motion characteristic of the scene, an architecture of the SIMD processor used in the motion compensated video coding.

22. The system of claim 21, wherein the matrix determination unit is further adapted to generate the pattern matrix based on a pixel budget, which is an estimate of number of pixels in the subset of pixels.

23. The system of claim 22, wherein the matrix determination unit further comprises:
a pattern matrix repository, storing a series of predetermined pattern matrices, each pattern matrix in the pattern matrix repository corresponding to the SIMD processor architecture used in the motion compensated video coding, the global motion characteristic of the scene, and the pixel budget; and

a pattern matrix selection unit, selecting the pattern matrix from the pattern matrix repository.

24. The system of any one of claims 15, 16, 17, 18, 21, 22, 23, wherein the metric computation unit comprises one of a Sum of Absolute Differences computing unit, applying the Sum of Absolute Differences function on values of the characteristic in the subset of pixels, or a Mean Squared Error computing unit, applying the Mean Squared Error function on values of the characteristic in the subset of pixels.

25. The system of any one of claims 15, 16, 17, 18, 21, 22, 23, wherein positions of said one or more groups of successive pixels are selected so that as to maximize a span of block coverage.

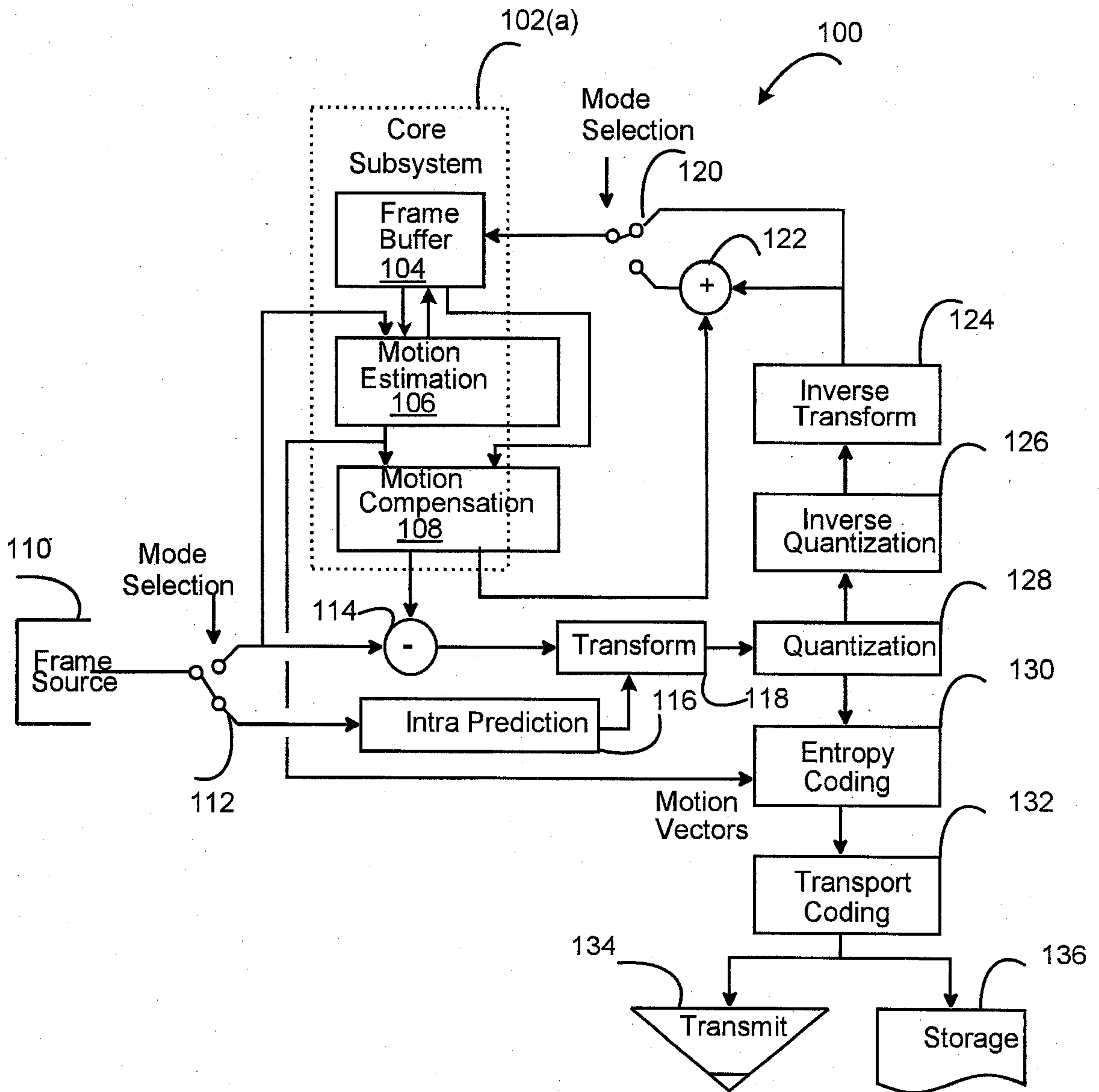


Figure 1 (Prior Art)

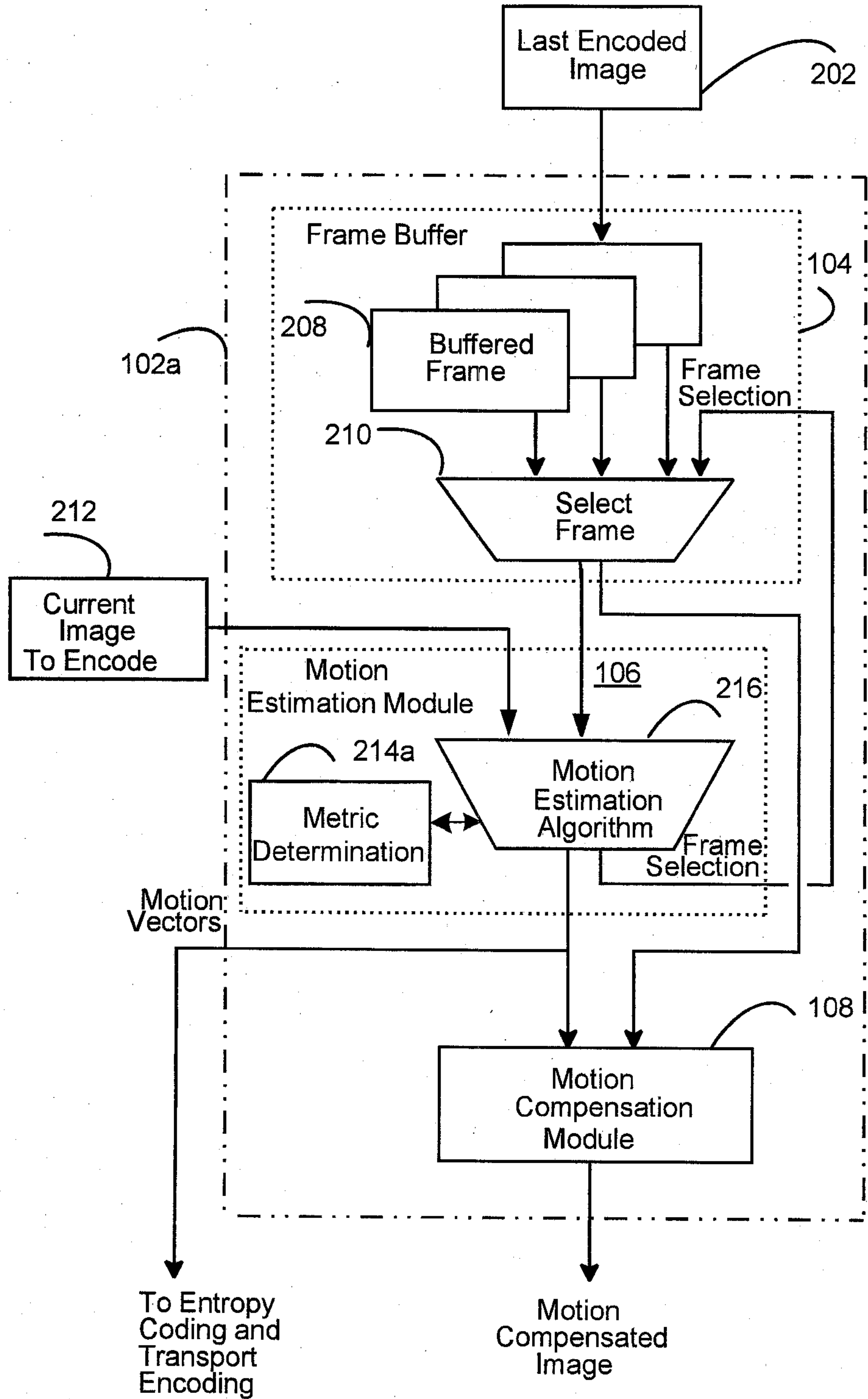


Figure 2 (Prior Art)

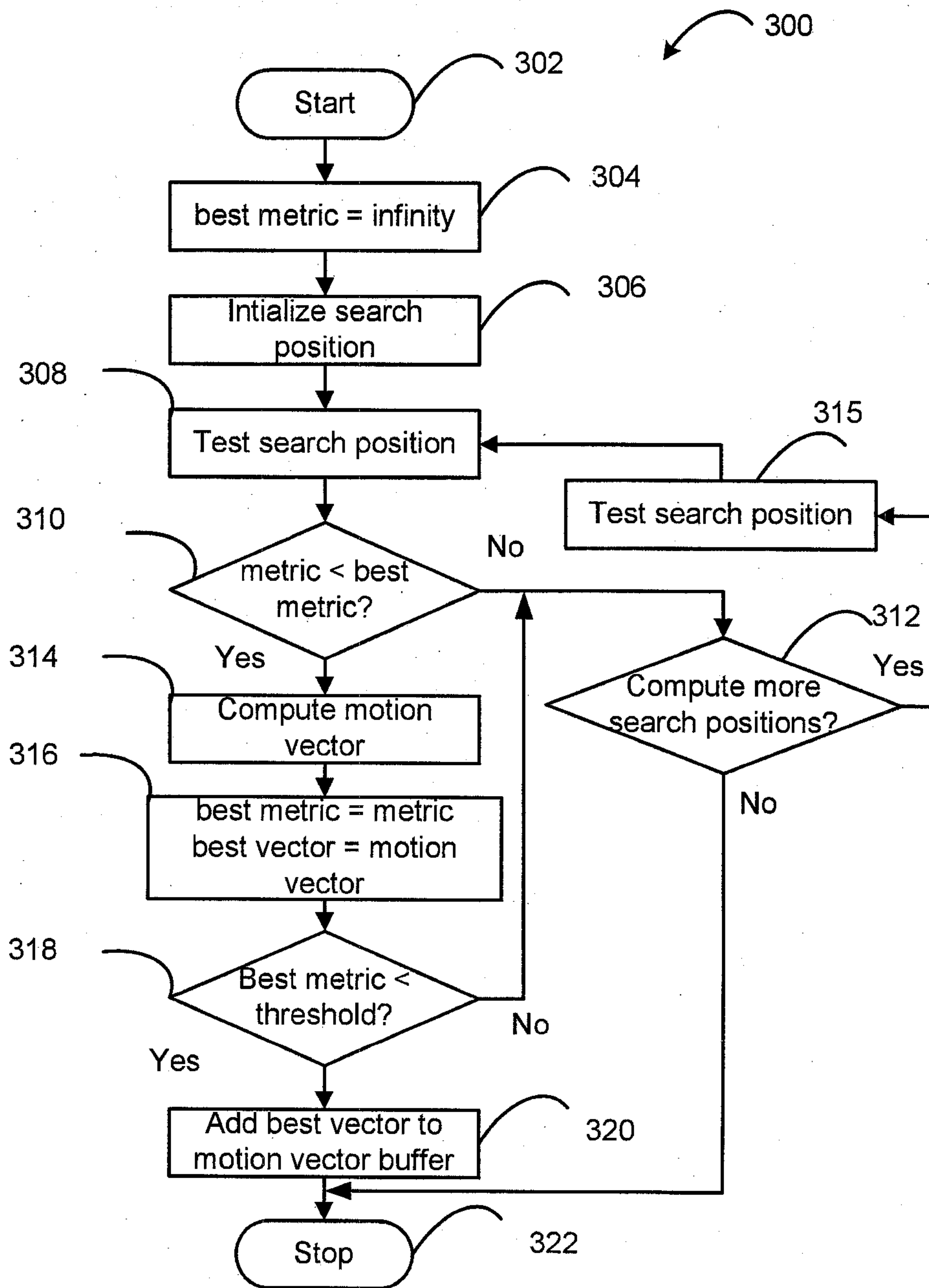


Figure 3 (Prior Art)

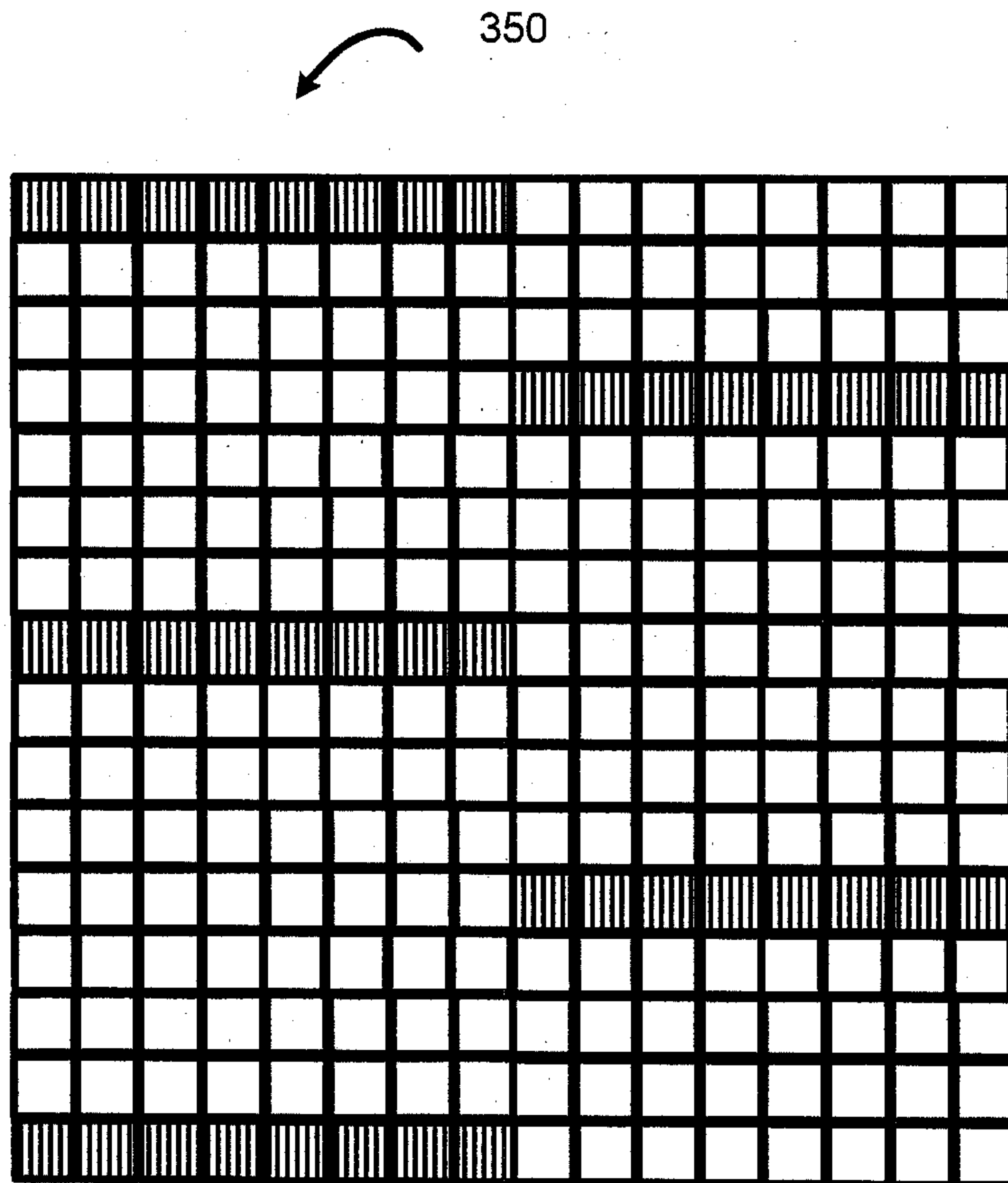


Figure 4 (a)

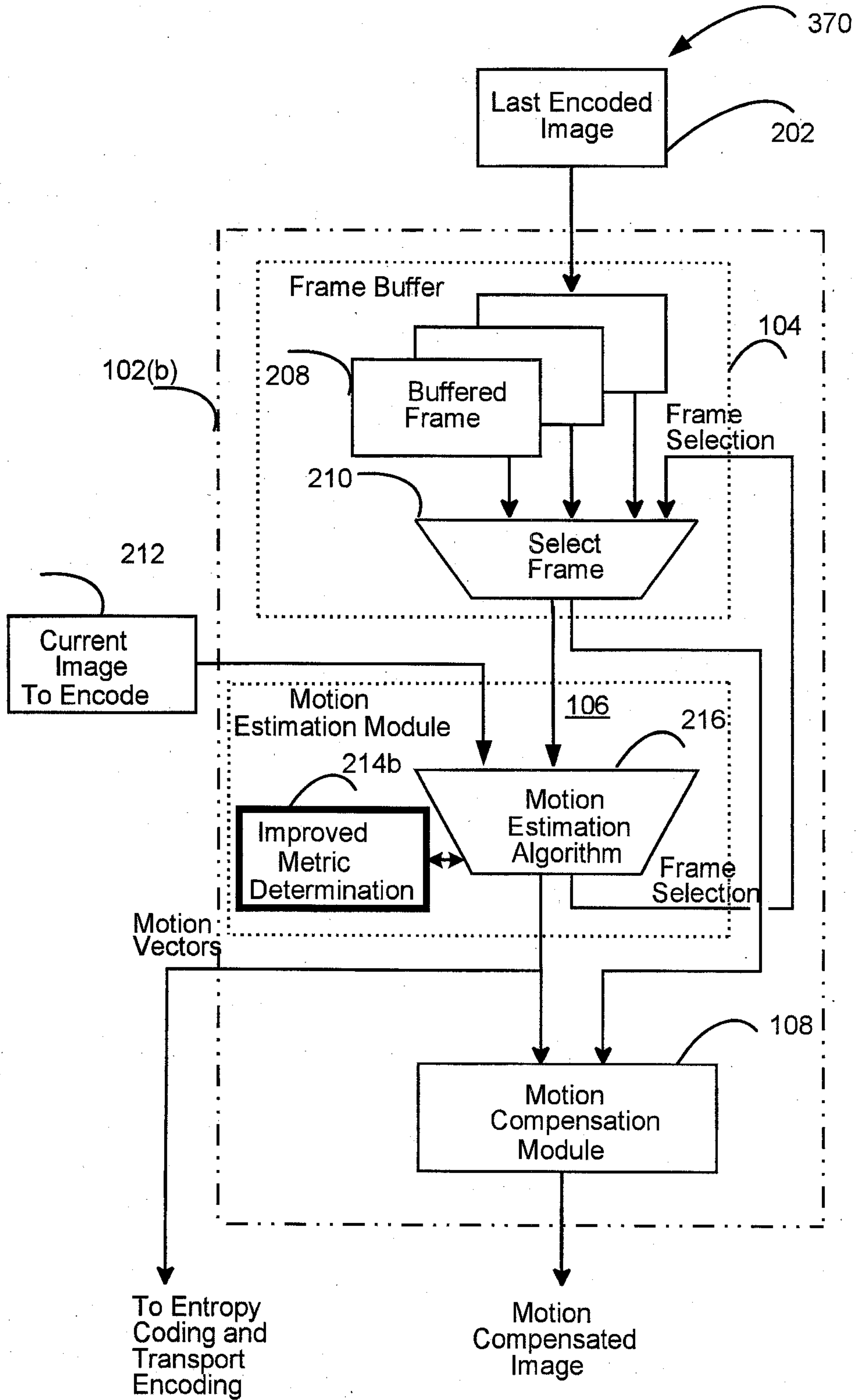


Figure 4(b)

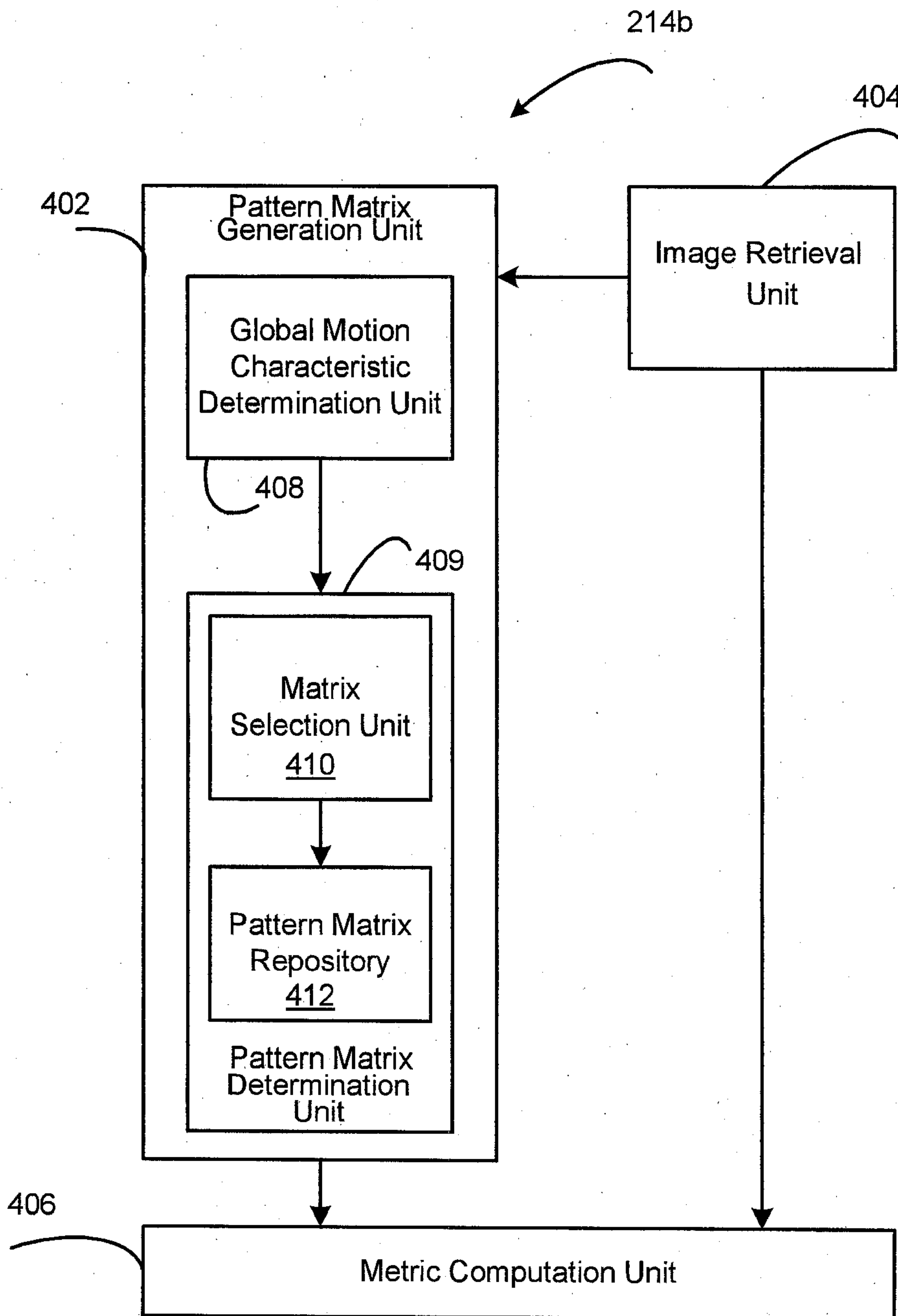


Figure 4 (c)

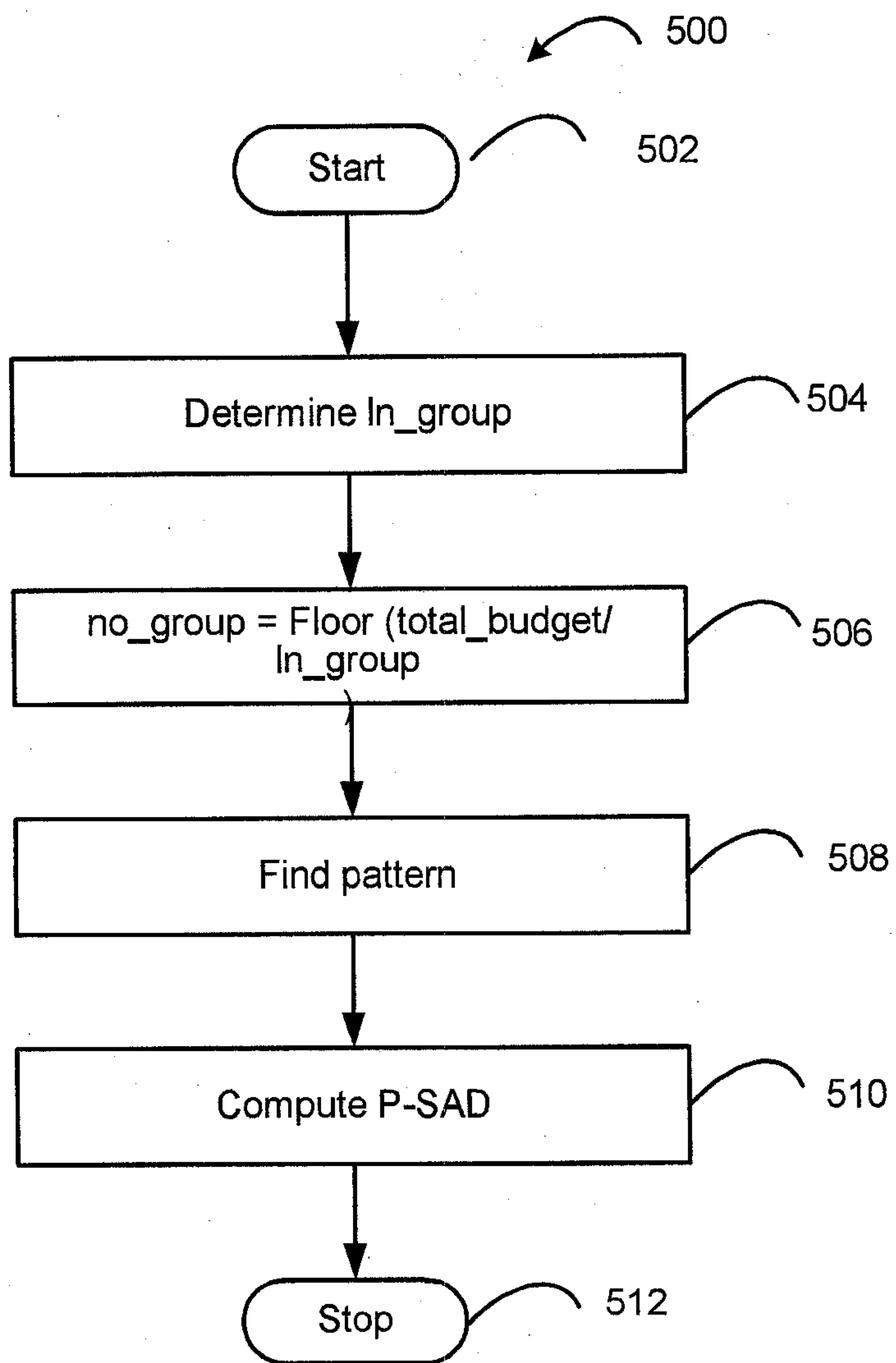


Figure 5

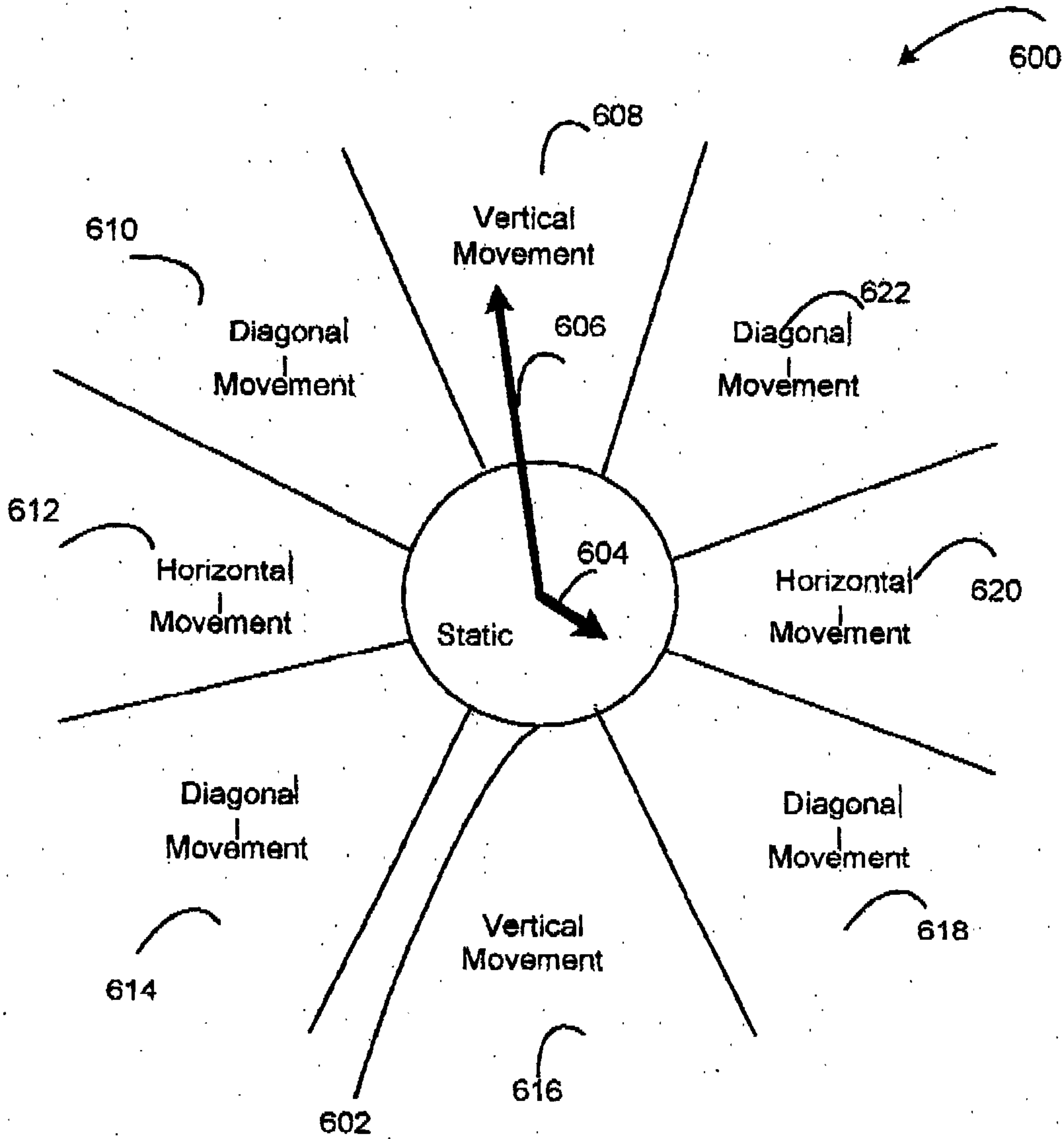


Figure 6

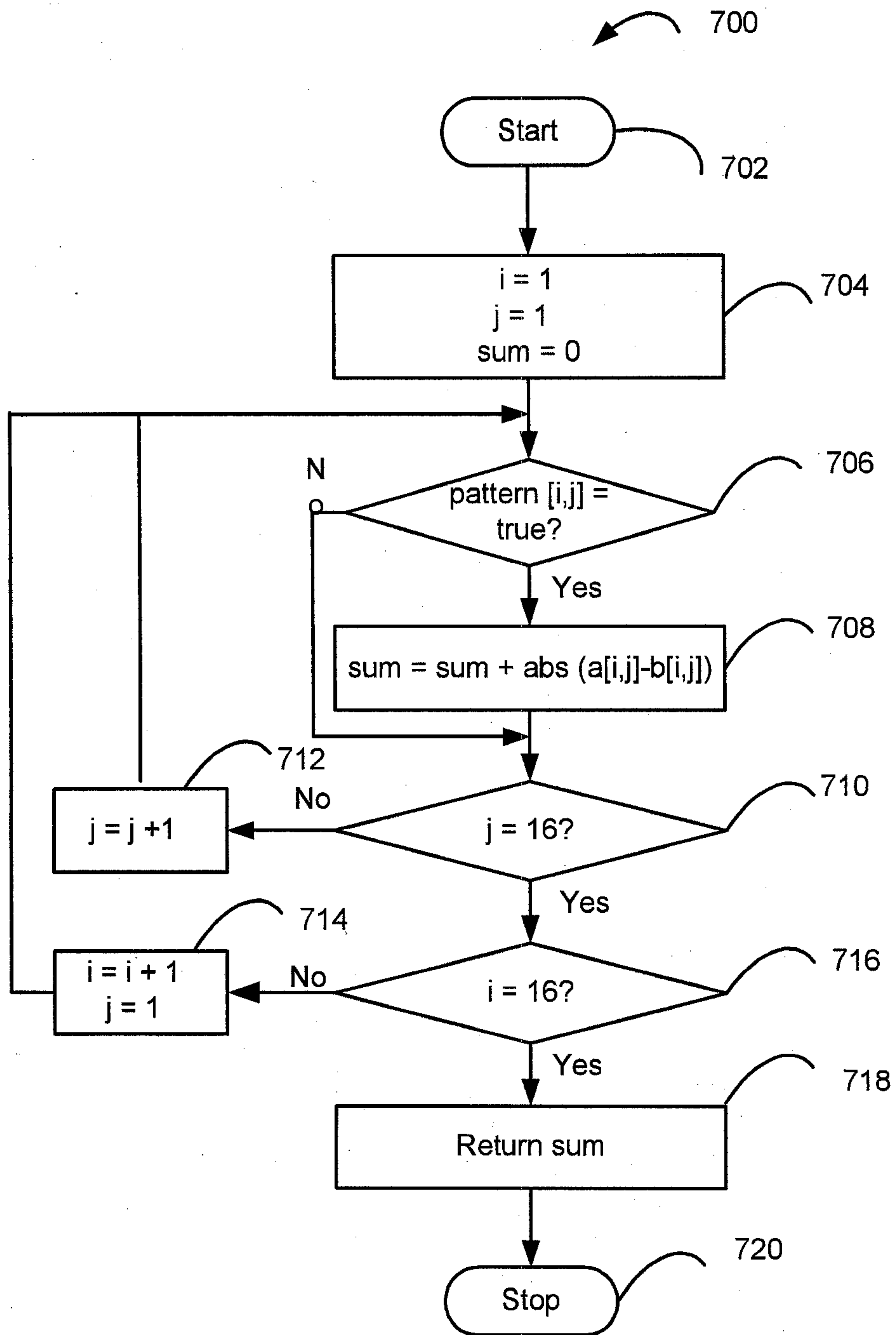


Figure 7

