(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 2 383 992 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
02.11.2011 Bulletin 2011/44

(21) Application number: **11163706.2**

(22) Date of filing: **26.04.2011**

(51) Int Cl.:
H04N 7/01 $^{(2006.01)}$ ***H04N 7/26*** $^{(2006.01)}$
H04N 5/44 $^{(2011.01)}$ ***H04N 5/14*** $^{(2006.01)}$

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME

(30) Priority: **30.04.2010 TR 201003470**

(71) Applicant: **Vestel Elektronik Sanayi ve Ticaret A.S.**
45030 Manisa (TR)

(72) Inventors:
- **Gunyel, Bertan**
  45030, Manisa (TR)
- **Alatan, Abdullah Aydin**
  45030, Manisa (TR)

(74) Representative: **Cayli, Hülya**
Paragon Consultancy Incorporated
Koza Sokak No: 63/2
GOP 06540 Ankara (TR)

(54) Method and apparatus for the detection and classification of occlusion regions

(57) In frame rate up-conversion for video signals interpolated frames are inserted within the video stream to increase the frame rate for display. Halo effects, observed as ghost replicas, can occur in occlusion regions around the boundaries of moving objects. To enable the occlusion regions to be compensated, they are detected by a method utilizing forward, backward and bidirectional motion vectors for each of a first frame (100), a second frame (104) and an interpolated frame (102). Occlusion regions (106,108) are located by comparing bidirectional motion vectors with forward or backward motion vectors to find inconsistencies. Then blocks of pixels having bi directional motion vectors (112,114) can be compensated unidirectionally by using the pixel information (220,224) in one of the original frames (100,102).

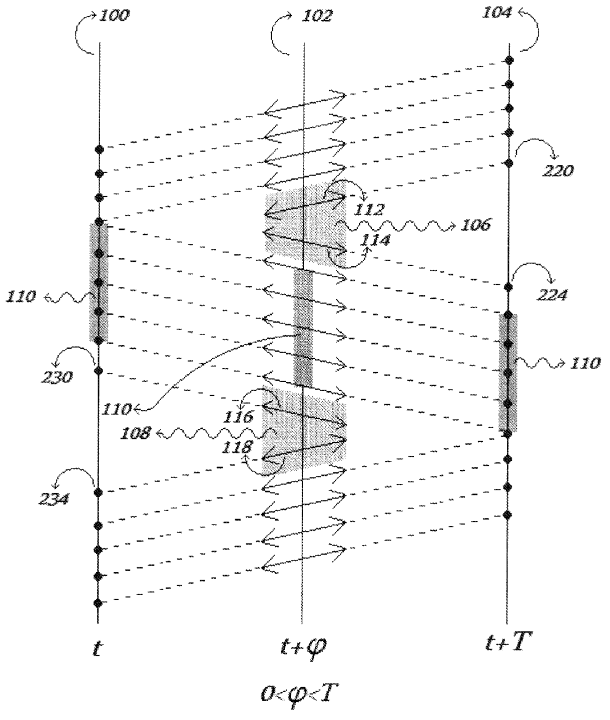


Figure 2

Printed by Jouve, 75001 PARIS (FR)

Description

Technical Field

[0001] The present invention relates to a method and apparatus for the detection and classification of occlusion regions.

Prior Art

[0002] A video is made up of plural frames, each frame being a still image, the effect of showing consecutive frames being to give the impression of movement. The frame rate of videos is generally sufficiently high such that a viewer cannot perceive the individual frames but rather perceives a continuous moving image.

[0003] On some television displays, it is desirable to increase the frame rate of displayed video from its original frame rate. This can be due to higher frame rate requirement for display in accordance with a different standard from that at which the video was captured.

[0004] In addition, many of the latest high-performance television sets, particularly large screen and wide screen (16: 9 format) versions, use a 100Hz screen refresh rate instead of the conventional 50Hz refresh rate. This is primarily to avoid screen flicker and motion blur, which becomes more perceptible and annoying to the viewer as the screen size increases. However, standard television broadcast transmissions only contain 50 frames per second. Hence, the frame rate of the transmitted content should be adjusted for, for example, the new LCD televisions which have refresh rates of 100Hz, 120Hz or 200Hz.

[0005] The increase in the frame rate is called frame rate up conversion. For frame rate up conversion, new frames are inserted between existing frames of the video content. Three straightforward approaches for frame rate up conversion are referred to as "black frame insertion", "frame repetition" and "frame averaging". Whilst these techniques are computationally relatively simple, they can degrade visual performance. For example, black frame insertion reduces the average intensity of the video content, whereas frame repetition and frame averaging cause "motion judder" and "motion blur". Motion compensated frame interpolation is the preferred solution for interpolation to avoid the motion blur, which is caused by the motion persisting function of the human visual system, the long response time of the crystal cells and the hold-type characteristic of panel displays.

[0006] However, with motion compensated frame rate up conversion, halo effects, observed as "ghost replicas" in occlusion regions around the boundaries of moving objects, arise. Several approaches have been attempted to cope with the halo problem. Since the luminance information in occlusion regions is available only in one of two consecutive frames, it has been suggested that more than two frames be used both in motion estimation and in motion compensation.

[0007] However, it is often necessary to limit the number of frames to two. In that circumstance an occlusion handling algorithm is used as described in US 6219436. The method proposed in this US patent specification makes use of the fact that forward and backward matching errors cannot be simultaneously small in occlusion regions due to lack of corresponding intensity information in consecutive frames. Hence, the occlusion regions are detected according to the ratio between forward and backward matching errors and cover/uncover classification is performed according to the relative magnitudes of these errors.

[0008] US 6487313 patent application proposes an occlusion detector and classifier algorithm which detects occlusion regions according to sharp changes in the motion vector field. The classification of the occlusion regions is performed according to relative positions of corresponding vectors in space with respect to each other.

[0009] WO-A-97/04599 patent application detects occluded regions based on the idea that the vector trajectories of non-occluded blocks should be similar in consecutive motion fields. Discontinuity of trajectories is used to label occlusion blocks.

[0010] These techniques have not been found to be as robust and reliable as is required in practice.

Brief Description of the Invention

[0011] Present invention provides a method for detecting occlusion regions on an interpolated frame to be located temporally between first and second frames of a sequence of video frames, the method utilizing forward, bidirectional and backward motion vectors estimated for each of the first, interpolated, and second frames, characterized in that the method comprises comparing at least one bidirectional motion vector with either a forward or a backward motion vector to find an inconsistency and thereby identify the location on the interpolated frame of an occlusion region.

[0012] Embodiments of the method enable occlusion regions to be detected. In an embodiment, to find inconsistencies between a bidirectional motion vector and either a forward or a backward motion vector, the sum of the absolute differences of the motion vector components between the bidirectional motion vector and either a forward or a backward motion vector is compared with a predetermined threshold.

[0013] The method preferably further comprises investigating a plurality of regions of the interpolated frame and identifying regions at which inconsistencies are found as occlusion regions, and other regions where there are no inconsistencies as non-occluded regions.

[0014] In another embodiment, the method further comprises classifying the occlusion regions located as covered, or uncovered.

[0015] The identification of the regions of the interpolated frame, and the classification of the located occlusion regions, enables different interpolation strategies to be employed for each type of region identified. Generally, non-occluded regions will be interpolated from both first and second frames. The information to interpolate an uncovered region will be taken from the next, second frame, whilst the information to interpolate a covered region will be taken from the previous, first frame.

[0016] Preferably, to classify the located occlusion regions the weighted sum of the differences of x and y components of the motion vector of a pixel, or a block of pixels, is compared with a predetermined classification threshold.

[0017] Preferably, a plurality of decision steps is included in the calculation of the weighted sum of the vector component differences. To provide for good classification performance without adding unnecessary computational complexity, four decision steps might be included.

[0018] Once the occlusion regions have been detected and classified, they can be interpolated. Preferably the occlusion regions are interpolated according to their classification.

[0019] The present invention also extends to a method of frame interpolation for a sequence of video frames comprising locating an interpolated frame temporally between first and second frames of the sequence, the method comprising;

- investigating a plurality of regions of the interpolated frame to detect occlusion regions on the interpolated frame by utilizing forward, bidirectional and backward motion vectors estimated for each of the first, interpolated, and second frames, and comparing at least one bidirectional motion vector with either a forward or a backward motion vector;
- identifying regions at which inconsistencies are found by the comparison as occlusion regions, and identifying other regions as non-occluded regions;
- classifying the occlusion regions located as covered, or uncovered; and
- interpolating the interpolated frame according to the classification of covered, uncovered and non-occluded regions.

[0020] The frame interpolation method of an embodiment of the invention may use any of the detection and/or classification techniques defined above. Similarly, the defined detection and classification methods may be combined with any of the following interpolation methods. Alternatively, any of the individual techniques described may be used alone.

[0021] In an embodiment, methods as defined above may comprise interpolating regions identified as non-occluded utilizing information from both of the first and second frames;

- interpolating occlusion regions classified as covered by utilizing information from the previous, first frame; and
- interpolating occlusion regions classified as uncovered, by utilizing information from the next, second frame.

[0022] The information used for the interpolation may be pixel information, for example, the intensity values of individual pixels and/or blocks of pixels.

[0023] In a preferred embodiment, the interpolation or compensation of a non-occluded region in the interpolated frame comprises making a weighted summation of corresponding pixels in the first frame and the second frame.

[0024] A corresponding pixel in the first frame may be found at a spatial position which is determined by the location of the non-occluded region in the interpolated frame using the corresponding bidirectional motion vector. Similarly, the corresponding pixel in the second frame may be found at a spatial position which is determined by the location of the non-occluded region in the interpolated frame using the corresponding bidirectional motion vector.

[0025] Preferably, the coefficients for the weighted summation of the corresponding pixels from the first frame and the second frame are calculated according to the ratio of the temporal position of the interpolated frame to the time between the first and second frames less the temporal position of the interpolated frame.

[0026] In an embodiment, the covered and uncovered occlusion regions are interpolated using the weighted summation of unidirectional motion compensated responses of neighboring vectors. For example, the unidirectional motion compensated responses of neighboring vectors are taken from the first or the second frame as appropriate, at a spatial position which is determined by the position in the interpolated frame, the corresponding bidirectional motion vector of the corresponding neighboring location and the above defined temporal ratio.

[0027] Preferably, the weights utilized in the weighted summation should be normalized to give a sum of weights equal to 1.

[0028] Preferably, the method further comprises applying de-blocking filtering between neighboring blocks of pixels. The present invention also extends to apparatus for detecting occlusion regions on an interpolated frame to be located temporally between first and second frames of a sequence of video frames, the apparatus comprising

- occlusion detection means arranged to receive forward, bidirectional and backward motion vectors estimated for each of the first, interpolated, and second frames,
- the occlusion detection means comprising comparator means for comparing at least one bidirectional motion vector with either a forward or a backward motion vector to find an inconsistency,
- the comparator means being arranged to output an identification signal when an inconsistency is found.

[0029] Preferably, the apparatus further comprises occlusion classifier means arranged to classify the occlusion regions located as covered, or uncovered.

[0030] In an embodiment, the occlusion classifier means comprises computing means arranged to compute the x and y component differences of applied motion vectors, summing means for producing the weighted sum of the component differences, and comparator means for comparing the computed output of the summing means with a predetermined classification threshold.

[0031] The occlusion classifier means further comprises control logic arranged to cycle the computation through the computing means and summing means for a number of passes before the computed output is applied to the comparator means.

[0032] For example, the number of passes is preferably four. Apparatus as defined above will be incorporated in apparatus for undertaking frame interpolation. For example, apparatus as defined may be incorporated within apparatus, such as television receivers and set top boxes, for receiving and displaying audio and video content.

Object of the Invention

[0033] The object of the present invention is to provide a method for occlusion detection that is to be used to process an interpolated frame in a video sequence.

[0034] Another object of present invention is to classify occlusions and process an interpolated frame in accordance with the said classification.

[0035] Another object of the present invention is to propose a method wherein de-blocking filtering between neighboring blocks of pixels of an interpolated frame is employed in order to prevent blocking artifacts.

[0036] Another object of the present invention is to provide an apparatus for occlusion detection on an interpolated frame.

Brief Description of the Drawings

[0037]

Figure 1 shows forward, backward and bidirectional motion vector fields.
Figure 2 shows the interpolation strategy to be applied to occluded and non-occluded regions.
Figure 3 shows a candidate vector selection for block matching.
Figure 4 shows a de-blocking filter structure for a vertical edge between two 4x4 blocks.
Figure 5 shows a de-blocking filter structure for a horizontal edge between two 4x4 blocks.
Figure 6 shows one dimensional kernel positions for the filtering of a first row around a vertical edge.
Figure 7 shows the one dimensional kernel positions for the filtering of a first column around a horizontal edge.
Figure 8 shows an apparatus for frame interpolation.
Figure 9 shows a block diagram of an occlusion detector for an apparatus as shown in Figure 8.
Figure 10 shows a block diagram of an occlusion classification unit for use in an apparatus as shown in Figure 8.
Figure 11 shows a block diagram of a frame interpolation unit as used in an apparatus as shown in Figure 8.
Figure 12 shows a block diagram of an adaptive de-blocking filtering unit for use in an apparatus as shown in Figure8.

[0038] The reference numbers and abbreviations as used in figures may possess the following meanings;

Foreground object	(110)
A first original frame	(100)
A second original frame	(104)
An interpolated frame	(102)
Occlusion region	(106, 108)
Pixels or blocks of pixels having bidirectional motion vector	(112, 114, 116, 118)
Backward motion vector	(120, 124, 128, 132)

(continued)

Forward motion vector	(122, 126, 130, 134)
Pixel information for uncovered region	(220, 224)
Pixel information for covered region	(230, 234)
Pixel (or block of pixels) to be compensated	(300)
Neighboring pixels (or blocks of pixels) to a pixel to be compensated	(301 - 309)
A first block of pixels to be deblocked for a vertical edge	(401 - 416)
A second block of pixels to be deblocked for a vertical edge	(421 - 436)
Boundary between the blocks of pixels to be deblocked for a vertical edge	(400)
Effected pixels by deblocking for a vertical edge	(403, 404, 407,408, 411, 412, 415, 416, 421, 422, 425, 426, 429, 430, 433 and 434)
A first block of pixels to be deblocked for a horizontal edge	(501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515 and 516)
A second block of pixels to be deblocked for a horizontal edge	(521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535 and 536)
Effected pixels by deblocking for a horizontal edge	(509, 510, 511, 512, 513, 514, 515, 516, 521, 522, 523, 524, 525, 526, 527 and 528)
Forward Motion Estimation Unit	MEFwd
Bidirectional Motion Estimation Unit	MEBid
Backward Motion Estimation Unit	MEBwd

Detailed Description of the Invention

[0039] Embodiments of present invention are to be utilized in methods for frame interpolation for a sequence of video frames. Thus, it is required to add interpolated frames to the sequence, each interpolated frame being located temporally between a first frame and a second frame of the sequence. Embodiments of the invention use forward, bidirectional and backward motion vectors estimated for each of the first, interpolated, and second frames in a sequence as shown in Figure 1.

[0040] The manner in which the motion vectors are determined or estimated is not significant. Furthermore, any appropriate motion estimation unit may be utilized. The estimated motion vector fields used are intended to provide the 2-dimensional projection of the real 3-dimensional motion within the scene.

[0041] Figure 1 schematically shows the movements of a foreground object (110) and the movement of the background together with the corresponding motion vector fields in a hypothetical scene. A first original frame, *F(t)*, is labeled (100), and a second original frame, *F(t+T)*, is labeled (104). A new frame (102) is to be interpolated at the temporal position $(t+\varphi)$, where $0<\varphi<T$. The foreground object (110), is moving in a downward direction with a velocity of $\vec{D}_{fg}$, whereas the background moves upwards with the velocity $\vec{D}_{bg}$. The regions (106) and (108) are occlusion regions. An occlusion region is an area of an image which is visible in a sequence of frames but which is not visible in either the first frame or the second frame. The arrows originating from the first frame (100) are the forward motion vectors which are estimated by finding a match in the second frame (104) for every pixel or block of pixels defined on first frame (100). Likewise, the arrows having their origins on second frame (104) represent the backward motion vectors which are calculated by matching pixels or blocks of pixels on first frame (100) for every pixel or block of pixels defined on second frame (104). Bidirectional motion vectors are estimated by finding one match on each of original frames (100) and (104) which are located on the same spatio-temporal line with the particular pixel or block of pixels on the interpolated frame (102). These bidirectional motion vectors are represented by the two sided arrows located on the interpolated frame (102).

[0042] It can be seen in Figure 1 that the bidirectional vectors, except for the ones located in the occlusion regions, are consistent with the corresponding forward and backward motion vectors, which are located at positions pointed by the particular bidirectional motion vector. The bidirectional motion vector of pixels or blocks of pixels (112), which is located in the occlusion region (106), is inconsistent with at least one of the corresponding backward and forward motion vectors, (120) and (122). Similarly, the bidirectional motion vector of pixels or blocks of pixels (114) which is also in occlusion region (106) is not similar to at least one of the corresponding backward and forward motion vectors (124) and (126). In the occlusion region (108), the bidirectional motion vector of pixels or blocks of pixels (116) is not consistent with at least one of the vectors (128) and (130), whereas bidirectional motion vector of pixels or blocks of pixels (118) is dissimilar to at least one of vectors (132) and (134).

[0043] Accordingly, denoting the particular bidirectional motion vector $\vec{D}_{bid}$ ($\vec{x}$,t+φ) at the spatial position $\vec{x}$, simply as $\vec{D}_{\mathrm{bid}}$, occlusion regions can be detected as set out in relation (1).

$$\vec{x} \in R_{occlusion}\ if \begin{cases} Diff\left(\vec{D}_{bid}, \vec{D}_{forward}\left(\vec{x} - \lambda \cdot \left(\dfrac{\varphi}{T}\right) \cdot \vec{D}_{bid}, t\right)\right) \geq thr_{occ} \\ OR \\ Diff\left(\vec{D}_{bid}, \vec{D}_{backward}\left(\vec{x} + \lambda \cdot \left(1 - \dfrac{\varphi}{T}\right) \cdot \vec{D}_{bid}, t + T\right)\right) \geq thr_{occ} \end{cases} \tag{1}$$

[0044] In the relation (1), $R_{occlusion}$ is the set of occlusion regions in the interpolated frame and thr_{occ} is a predetermined threshold. The parameter λ is a scaling factor for enhancing the robustness of the detection algorithm. The function *Diff* (.) is defined in equation (2), where D^x and D^y denote the x and *y* components of the vectors, respectively.

$$Diff\left(\vec{D}_1, \vec{D}_2\right) = \left|D_1^x - D_2^x\right| + \left|D_1^y - D_2^y\right| \tag{2}$$

[0045] Thus, and as set out in relation (1) and equation (2), a bidirectional motion vector is compared with the forward and backward motion vectors, and an inconsistency is found if the difference exceeds a predetermined threshold. As shown in equation (2), it is the sum of the absolute differences of the motion vector components of the vectors which are compared.

[0046] When an occlusion region is located, it is classified. It is necessary to determine whether any detected occlusion region is covered, as occlusion region (108) in Figure 1, or is uncovered, as occlusion region (106) in Figure 1. The detection and classification determines how occluded and non-occluded regions are rendered, or interpolated, in the interpolated frame (102). Generally, the necessary information for the interpolated frame (102) is taken from the next, or second, frame (104) if the occlusion region is uncovered, as in occlusion region (106). The information is taken from the previous, or first, frame (100), if the occlusion region is covered, as in occlusion region (108).

[0047] Regions which are not identified as occlusion regions, that is, regions where there is no inconsistency between a bidirectional motion vector and corresponding forward and backward motion vectors, are identified as non-occluded regions. For such non-occluded regions the necessary information for the interpolated frame (102) is taken from both of the first and second frames.

[0048] When a detected occlusion region can not be classified accurately, it can be left as an unclassified or undecided occlusion region. For these regions, information for the interpolated frame may be performed bidirectional, that is, can be taken from both of the first and second frames.

[0049] For the occlusion classification algorithm, the origin of the image plane is assumed to be the top left corner of the image frame. Hence, the positive x and *y* directions point from left to right and from top to bottom, respectively.

[0050] The classification method analyses x and y components of the vector pairs located at different positions in the neighborhood of the particular pixel or block of pixels in the interpolated frame *F(t*+φ*)*, (102). The classification is based on the weighted sum of the vector component differences.

[0051] Denoting x and *y* components of the motion vector of a pixel or block of pixels with $D_x(x,y)$ and $D_y(x,y)$, where (x,y) represents the coordinates of the particular pixel or indices of the particular block, the weighted sum of the vector component differences, $\Psi_{x,y}$, is defined as in equation (3).

$$\psi_{x,y} = \sum_{i=0}^{N-1} \gamma_i \left[c_x D_x^{diff}(x,y) + c_y D_y^{diff}(x,y)\right] \tag{3}$$

$$D_x^{diff}(x,y) = D_x(x - d_i, y) - D_x(x + d_i, y) \tag{4}$$

$$D_y^{diff}(x,y) = D_y(x, y-d_i) - D_y(x, y+d_i) \tag{5}$$

[0052] The parameter, d_i, controlling the vector pairs used in the analysis should be different for each of the N decision steps. The coefficient γ_i determines the dominancy of each vector pair in the classification.

[0053] In an environment, where the horizontal component of motion is more dominant than the vertical component, the x component differences of the vector pairs are more informative than the *y* component differences. Likewise, the *y* component differences are more valuable in an environment where the vertical motion is more dominant than the horizontal motion. Hence, the coefficients c_x and c_y should be calculated accordingly.

$$c_x = \frac{|D_x(x,y)|}{|D_x(x,y)| + |D_y(x,y)|} \tag{6}$$

$$c_y = \frac{|D_y(x,y)|}{|D_x(x,y)| + |D_y(x,y)|} \tag{7}$$

[0054] Preferably, the number of decision steps, N, is set to be 4 which is an optimal choice between the computational complexity and the classification performance. Accordingly, the parameters d_i and γ_i are formulized as in equations (8) and (9).

$$d_i = 1 + 2i, \quad i = 0,1,2,3 \tag{8}$$

$$\gamma_i = \frac{1+2i}{\sum_{i=0}^{3}(1+2i)}, \quad i = 0,1,2,3 \tag{9}$$

[0055] After the weighted sum of vector component differences, $\Psi_{x,y}$, has been calculated, it is compared with a positive classification threshold in order to make the classification.

$$(x,y) \in \begin{cases} Uncover, & \psi_{x,y} < -thr_{classification} \\ Cover, & \psi_{x,y} > thr_{classification} \\ Undecided, & otherwise \end{cases} \tag{10}$$

[0056] As set out in relation (10), different interpolation strategies will be used for regions categorized differently.

[0057] Figure 2 schematically illustrates the interpolation strategy to be applied to covered, uncovered and non-occluded regions. The non-occluded regions of the interpolated frame *F*(*t*+φ) are compensated by using bidirectional pixel averaging, since the corresponding intensity information is present in both frames *F*(*t*) and *F*(*t*+*T*). The occlusion regions need to be interpolated with a unidirectional scheme as the corresponding intensity information is only present in one of *F*(*t*) or *F*(*t*+*T*). The correct pixel information for the uncovered region (106) is obtained from the next, or second frame (104) *F*(*t*+*T*). Hence, the pixels or blocks of pixels having bidirectional motion vectors (112) and (114) of the uncovered region (106) are compensated unidirectionally by using the pixel information labeled (220) and (224), respectively. The correct intensity information for the covered region (108) is taken from the previous, or first frame (100), *F* (*t*). The pixels or blocks of pixels having bidirectional motion vector (116) and (118) of the covered region (108) are compensated by the pixel information (230) and (234) of frame (100), respectively.

[0058] Since the bidirectional block matching might not yield correct motion vectors in covered and uncovered regions, the unidirectional interpolation from the correct frame might still create visual artifacts. This can be observed from the motion vectors of the covered and uncovered regions (106), (108) in Figure 2. The pixels or blocks of pixels having the bidirectional motion vectors (112) and (114) and which are spatially adjacent, are interpolated by using pixel information (220) and (224) which are spatially unrelated. This can cause inconsistency in the visual performance. Similar artifacts may be observed in the compensation of the adjacent pixels or blocks of pixels having bidirectional motion vector (116) and (118) by information from the non-adjacent pixel information (230) and (234). In order to suppress such artifacts, a weighted compensation is utilized for the occlusion regions.

[0059] The unidirectional responses of a bidirectional motion vector $\vec{D}_{bid}$ for a pixel located at $\vec{X}$= (*x*,*y*) are illustrated in equations (11) and (12) for covered and uncovered regions, respectively.

$$R_{cover}\left(\vec{X},\vec{D}_{bid},t+\varphi\right)=F\left(\vec{X}-\left(\frac{\varphi}{T}\right)\cdot\vec{D}_{bid},t\right) \qquad (11)$$

$$R_{uncover}\left(\vec{X},\vec{D}_{bid},t+\varphi\right)=F\left(\vec{X}+\left(1-\frac{\varphi}{T}\right)\cdot\vec{D}_{bid},t+T\right) \qquad (12)$$

[0060] The weighted compensation result is obtained by the weighted summation of the compensation responses of a number of candidate vectors.

$$I_{cover}\left(\vec{X},t+\varphi\right)=\frac{\sum_{i=0}^{N-1} w_i\cdot R_{cover}\left(\vec{X},\vec{D}_i,t+\varphi\right)}{\sum_{i=0}^{N-1} w_i} \qquad (13)$$

$$I_{uncover}\left(\vec{X},t+\varphi\right)=\frac{\sum_{i=0}^{N-1} w_i\cdot R_{uncover}\left(\vec{X},\vec{D}_i,t+\varphi\right)}{\sum_{i=0}^{N-1} w_i} \qquad (14)$$

[0061] Figure 3 illustrates an example of candidate vector selection in which the number of candidate vectors N is 9. The pixel (or block of pixels) (300) is compensated by using the vectors of the neighboring pixels (or blocks of pixels) (301, 302, 303, 304, 305, 306, 307 and 308).

[0062] Denoting the particular bidirectional motion vector $\vec{D}_{bid}$ ($\vec{x}$,*t*+φ) at the spatial position $\vec{x}$, simply as $\vec{D}_{bid}$ the bidirectional compensation response is as defined in equation (15).

$$I_{bid}\left(\vec{X},t+\varphi\right)=\left(1-\frac{\varphi}{T}\right)\cdot F\left(\vec{X}-\left(\frac{\varphi}{T}\right)\cdot\vec{D}_{bid},t\right)+\left(\frac{\varphi}{T}\right)\cdot F\left(\vec{X}+\left(1-\frac{\varphi}{T}\right)\cdot\vec{D}_{bid},t+T\right) \quad (15)$$

[0063] The overall compensation strategy is illustrated in relation (16).

$$F\left(\vec{X},t+\varphi\right)=\begin{cases} I_{bid}\left(\vec{X},t+\varphi\right),\vec{X}\in R_{non-occluded} \\ I_{cover}\left(\vec{X},t+\varphi\right),\vec{X}\in R_{cover} \\ I_{uncover}\left(\vec{X},t+\varphi\right),\vec{X}\in R_{uncover} \end{cases} \quad (16)$$

[0064] In relation (16), $R_{non\text{-}occluded}$, R_{cover}, $R_{uncover}$ denote the set of pixels (or blocks of pixels), which are labeled as non-occluded, covered and uncovered, respectively.

[0065] As defined in relation (16), and as described above, the covered, uncovered and non-occluded pixels (or blocks of pixels) are compensated differently. For block-wise implementations, if some or all of these types appear in a neighborhood, apparent visual blocking artifacts can be observed.

[0066] In order to reduce the blocking artifacts in block-wise implementations, de-blocking filtering can be used and operated on the adaptive interpolation output. The filtering is applied along the block edge between two neighboring blocks, if

- one of the blocks is occluded and the other is non-occluded or,
- both of the blocks are occluded with different types of occlusion (cover or uncover).

[0067] Figure 4 schematically illustrates the de-blocking filter structure for a vertical edge between two 4x4 blocks. The pixels (401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415 and 416) constitute the first block of pixels which is subject to deblocking filtering, whereas the second block is formed of pixels (421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435 and 436). The boundary between the two blocks of pixels is labeled (400). The de-blocking filtering is effective over the pixels (403, 404, 407, 408, 411, 412, 415, 416, 421, 422, 425, 426, 429, 430, 433 and 434).

[0068] The de-blocking filter structure for a horizontal edge between two 4x4 blocks is shown in Figure 5, where the first block includes the pixels (501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515 and 516), whereas the second block is formed of pixels (521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535 and 536). The boundary between the two blocks is labeled (500). The de-blocking filtering is effective on pixels (509, 510, 511, 512, 513, 514, 515, 516, 521, 522, 523, 524, 525, 526, 527 and 528).

[0069] The filtering is performed with a one dimensional kernel having a length of (*BlockWidth*+1) for vertical edges and (*BlockHeight* +1) for horizontal edges. The kernel length is adjusted to be an odd number for the self-dominancy of the filtered pixel.

[0070] Figure 6 schematically illustrates the one dimensional kernel positions for the filtering of the first row around a vertical edge (400). The two 4x4 blocks of pixels consist of the pixels (401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416) and (421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436), respectively. The pixels (403, 404, 421 and 422) are the filtered pixels on the first row, whereas the filtered pixels are labeled with (407, 408, 425 and 426) for the second row, with (411, 412, 429 and 430) for the third row and with (415, 416, 433, 434) in the fourth row for the 4x4 blocks of pixels in Figure 6. The filtering should be applied for the additional rows for block height dimensions greater than 4. The effective pixels for the filtering of a first pixel (403) are pixels (401, 402, 403, 404 and 421), as shown in (a), whereas pixels (402, 403, 404, 421 and 422) are effective for the filtering of a second pixel (404), as illustrated in (b). Similarly, (c) illustrates that pixels (403, 404, 421, 422 and 423) contribute for the filtering of a third pixel (421). Lastly, as depicted in (d), pixels (404, 421, 422, 423 and 424) are effective for the filtering of pixel (422). The effective pixels for the filtering of pixels in other rows are as follows:

- For 407 on Row 2: 405, 406, 407, 408, 425
- For 408 on Row 2: 406, 407, 408, 425, 426
- For 425 on Row 2: 407, 408, 425, 426, 427
- For 426 on Row 2: 408, 425, 426, 427, 428
- For 411 on Row 3: 409, 410, 411, 412, 429
- For 412 on Row 3: 410, 411, 412, 429, 430
- For 429 on Row 3: 411, 412, 429, 430, 431
- For 430 on Row 3: 412, 429, 430, 431, 432
- For 415 on Row 4: 413, 414, 415, 416, 433
- For 416 on Row 4: 414, 415, 416, 433, 434
- For 433 on Row 4: 415, 416, 433, 434, 435
- For 434 on Row 4: 416, 433, 434, 435, 436

[0071] Figure 7 schematically depicts the one dimensional kernel positions for the filtering of the first column around a horizontal edge (500). The two 4x4 blocks of pixels consist of the pixels (501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516) and (521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536), respectively. The pixels (509, 513, 521 and 525) are the filtered pixels on the first column, whereas the filtered pixels are labeled with (510, 514, 522, and 526) for the second row, with (511, 515, 523, 527) for the third column and with (512, 516, 524, 528) in the fourth column for the 4x4 blocks of pixels in Figure 7. The filtering should be applied for the additional columns for block width dimensions greater than 4. The effective pixels for the filtering are as follows:

- For 509 on Column 1: 501, 505, 509, 513, 521
- For 513 on Column 1: 505, 509, 513, 521, 525
- For 521 on Column 1: 509, 513, 521, 525, 529
- For 525 on Column 1: 513, 521, 525, 529, 533
- For 510 on Column 2: 502, 506, 510, 514, 522
- For 514 on Column 2: 506, 510, 514, 522, 526
- For 522 on Column 2: 510, 514, 522, 526, 530
- For 526 on Column 2: 514, 522, 526, 530, 534
- For 511 on Column 3: 503, 507, 511, 515, 523
- For 515 on Column 3: 507, 511, 515, 523, 527
- For 523 on Column 3: 511, 515, 523, 527, 531
- For 527 on Column 3: 515, 523, 527, 531, 535
- For 512 on Column 4: 504, 508, 512, 516, 524
- For 516 on Column 4: 508, 512, 516, 524, 528
- For 524 on Column 4: 512, 516, 524, 528, 532
- For 528 on Column 4: 516, 524, 528, 532, 536

[0072] The coefficients in the filtering window, and hence, the filter strength is adjusted adaptively according to the difference along the edge between the two blocks. The difference is calculated with one pixel depth on both sides along the edge to be filtered. The edge pixel depth for the edge difference calculation may be set to different values in different implementations. The edge difference is converted to pixel-wise basis via division with the edge length in pixels. The kernel coefficients are calculated as formulized in equation (17).

$$\varphi(k) = e^{-k^2/\sigma^2} \;,\; k = 0, \pm 1, \pm 2, \ldots \qquad (17)$$

[0073] In equation (17), σ^2 is the variance of the Gaussian formulation. After all coefficients are calculated, they are normalized to give a summation of 1.

[0074] The variance, σ^2, is set to be equal to the pixel-wise edge difference. Since the center coefficient is always set as 1 before normalization, the relative effectiveness of the other pixels increase with larger pixel-wise edge difference. Hence, stronger filtering is applied along edges with large difference values.

[0075] Figure 8 illustrates an apparatus for interpolating a sequence of video frames. As shown, a first frame (*F(t)*) and a second frame (*F(t+T)*) are applied to the motion estimation units (*MEFwd*), (*MEBid*) and (*MEBwd*), for the estimation of forward, bidirectional and backward motion vector fields. The estimated forward (*DFwd*), bidirectional *(DBid)* and backward *(DBwd)* motion vector fields are then applied to an occlusion detection unit (OD) to generate an occlusion map.

[0076] The occlusion detection unit (OD) investigates the estimated forward, bidirectional and backward motion vector fields to identify regions where there are inconsistencies, which are identified as occlusion regions, and regions where there are no inconsistencies, which are identified as non-occluded regions. The generated occlusion map is then processed by an occlusion classifier unit (OC) which classifies the occlusion regions as covered or non-covered. The output of the occlusion classifier unit is an occlusion classification map. The occlusion classification map is fed to an interpolation (INT) module, together with the first frame (F(t)) and the second frame (F$(t$+T$)$), for the generation of an initial output frame ($F'(t$+φ$)$). The last module of the apparatus is the adaptive filtering (FILT) unit, which receives the initial output frame ($F'(t$+φ$)$) and the occlusion classification map as input, and interpolates the final output of the system (F$(t$+φ$)$). The final output frame ($F(t$+ φ$)$) is the frame to be interpolated. The resulting interpolated sequence of frames is supplied to a display device to be displayed at an increased frame rate such as 100 Hz, 120 Hz or 200 Hz.

[0077] Figure 9 is a block diagram of an occlusion detector (OD) for use in an apparatus as shown in Figure 8. The occlusion detector (OD) has comparison units, (CMP1, CMP2), which each compare corresponding motion vectors and generate respective binary output signals (CMPOut1, CMPOut2). Specifically, forward motion vectors and bidirectional motion vectors are applied to a first comparator (CMP1) and its binary output signal, (CMPOut1), is set to 1 if the corresponding bidirectional motion vector, $\vec{D}_{bid}$, is inconsistent with the corresponding forward motion vector, $\vec{D}_f$. (CMPOut1) is set to 0 where there is no such inconsistency. Similarly, the backward motion vectors and bidirectional motion vectors are applied to a second comparator (CMP2) and its binary output signal, (CMPOut2), is set to 1 if the corresponding bidirectional motion vector, $\vec{D}_{bid}$, is inconsistent with the corresponding backward motion vector, $\vec{D}_b$, and is set to 0, otherwise. The binary output signals, (CMPOut1) and (CMPOut2) are then applied to a decision logic circuit so that a decision as to whether a particular region of the interpolated frame is occluded or non-occluded can be made. In the embodiment illustrated in Figure 9, the decision is a comparison utilizing a Boolean OR operation.

[0078] Figure 10 schematically shows a block diagram of the occlusion classification unit (OC) of the apparatus of Figure 8. The units horizontal component difference unit (DIFFH) and vertical component difference unit (DIFFV) compute horizontal and vertical vector component differences, respectively, according to the d_i parameter which is defined in equations (4) and (5), and whose value is determined according to the i^{th} pass in the multipass occlusion classification procedure. The weighted sum vector component differences (WSVCD) is computed with respect to the coefficients, c_x and c_y, which are determined according to the spatial characteristics of the particular motion vector under examination. As the next step, multi-step weighted sum of vector component differences (MSWSVCD) is calculated with respect to the coefficient, γ_i, which is defined in equation (3), and whose value is determined with respect to the i^{th} pass in the multi-step occlusion classification procedure. In the last stage, after the multi-step procedure is completed, multi-step weighted sum of vector component differences (MSWSVCD), which is represented by $\Psi_{x,y}$ in equation (3), is compared with respect to a scalar threshold in order to decide the occlusion type of the particular pixel or block of pixels as covered, uncovered, or undecided.

[0079] Figure 11 is a block diagram of a frame interpolation unit (INT) of the apparatus shown in Figure 8. The de-multiplexer (DeMux) activates the corresponding compensation unit while de-activating the others, according to the occlusion classification map. The de-multiplexer (DeMux) sets the corresponding enable signal to 1, while keeping the other enable signals at 0. Compensation is performed in only the enabled unit among bidirectional compensation, unidirectional covered region compensation and unidirectional uncovered region compensation. The output value for the disabled modules can be set to an appropriate value. At the final stage of the interpolation unit (INT), the corresponding compensation result is selected by the multiplexer (Mux), according to the occlusion classification map.

[0080] Figure 12 depicts a block diagram of an adaptive de-blocking filtering unit (FILT) as used in the apparatus of Figure 8. The filtering decision unit (FiltDec) decides whether there is need for de-blocking filtering between two neighboring blocks, according to the occlusion classification map, and generates enabling signals (Enable) for other units. The pixel-wise edge difference between the neighboring blocks is calculated in edge difference calculation unit (EdgeDiffCalc) and the filter coefficients are calculated with respect to the pixel-wise edge difference in filter coefficient calculation unit (CoeffCalc). At the last stage the filter kernel (Filt) is applied to the image frame.

[0081] The process aims first at localizing pixel-wise or block-wise occlusion regions in the frame to be interpolated, and classifying the detected occlusion regions as covered and uncovered. The interpolation of the new frame is performed with respect to the detected covered and uncovered regions. In the last stage, for block-wise implementations, the algorithm proposes an adaptive de-blocking filtering for reducing the possible visual artifacts between neighboring blocks of pixels.

[0082] It will be appreciated that the processes described herein may be implemented by software, firmware or hardware, or some combination thereof. Furthermore, variations in, or modifications to, the embodiments as described and illustrated may be made within the scope of the appended claims.

Claims

1. An occlusion detection method on regions of an interpolated frame (102), which is to be located temporally between a first frame and a second frame (100,104) of a sequence of video frames, said method utilizing forward, bidirectional and backward motion vectors estimated for each of said first, interpolated, and second frames, **characterized in that** said method comprises;

   - comparing at least one bidirectional motion vector with either a forward or a backward motion vector to find an inconsistency and
   - identifying the location on said interpolated frame of an occlusion region.

2. An inconsistency search method, wherein an inconsistency between a bidirectional motion vector and either a forward or a backward motion vector is searched, according to Claim 1, wherein, sum of absolute differences of a motion vector components between said bidirectional motion vector and either said forward or said backward motion vector is compared with a predetermined threshold.

3. A method according to Claim 1 or Claim 2, which further comprises investigating a plurality of regions of said interpolated frame and identifying regions at which said inconsistencies are found as occlusion regions, and other regions where there are no inconsistencies as non-occluded regions.

4. A method according to Claim 3, further comprising classifying occlusion regions located as covered or uncovered.

5. A classification method for said located occlusion regions according to Claim 4, wherein weighted sum of differences of x and y components of a motion vector of a pixel, or a block of pixels, is compared with a predetermined classification threshold.

6. A method according to Claim 5, further comprising including a plurality of decision steps in calculation of said weighted sum of the vector component differences.

7. A method according to Claims 4 to 6, further comprising interpolating said occlusion regions according to their classification.

8. A method of frame interpolation for a sequence of video frames comprising locating an interpolated frame (102) temporally between a first frame and a second frame (100,104) of said sequence, said method comprising;

   - investigating a plurality of regions of said interpolated frame for occlusion regions on said interpolated frame, utilizing forward, bidirectional and backward motion vectors estimated for each of said first, interpolated, and second frames, and comparing at least one bidirectional motion vector with either a forward or a backward motion vector;
   - identifying regions at which inconsistencies are found by said comparison as occlusion regions, and identifying other regions with no inconsistencies as non-occluded regions;
   - classifying said identified occlusion regions as covered, or uncovered; and
   - interpolating said interpolated frame according to said classification of covered, uncovered and non-occluded regions.

9. A method according to Claim 8, further comprising:

   - interpolating regions identified as non-occluded utilizing information from both of said first frame and said second frame;
   - interpolating said occlusion regions classified as covered by utilizing information from said first frame; and
   - interpolating occlusion regions classified as uncovered, by utilizing information said second frame.

10. A method according to Claim 9, wherein said covered occlusion regions and said uncovered occlusion regions are interpolated using weighted summation of unidirectional motion compensated responses of neighboring vectors.

11. A method according to Claims 8 to 10, further comprising applying de-blocking filtering between neighboring blocks of pixels.

12. Apparatus for detecting occlusion regions on an interpolated frame (102) to be located temporally between first and second frames (100,104) of a sequence of video frames, the apparatus comprising;

    - occlusion detection means arranged to receive forward, bidirectional and backward motion vectors estimated for each of a first frame, an interpolated frame, and a second frame,
    - occlusion detection means comprising comparator means for comparing for an inconsistency at least one bidirectional motion vector with either a forward or a backward motion vector,
    - comparator means being arranged to output an identification signal when an inconsistency is found.

13. Apparatus according to Claim 12, further comprising occlusion classifier means arranged to classify occlusion regions located as covered, or uncovered.

14. Apparatus according to Claim 13, wherein the occlusion classifier means comprises computing means arranged to compute x and y component differences of applied motion vectors, summing means for producing weighted sum of said component differences, and comparator means for comparing said computed output of said summing means with a predetermined classification threshold.

15. Apparatus according to Claim 14, wherein occlusion classifier means further comprises control logic arranged to cycle said computation through said computing means and summing means for a number of passes before the computed output is applied to said comparator means.

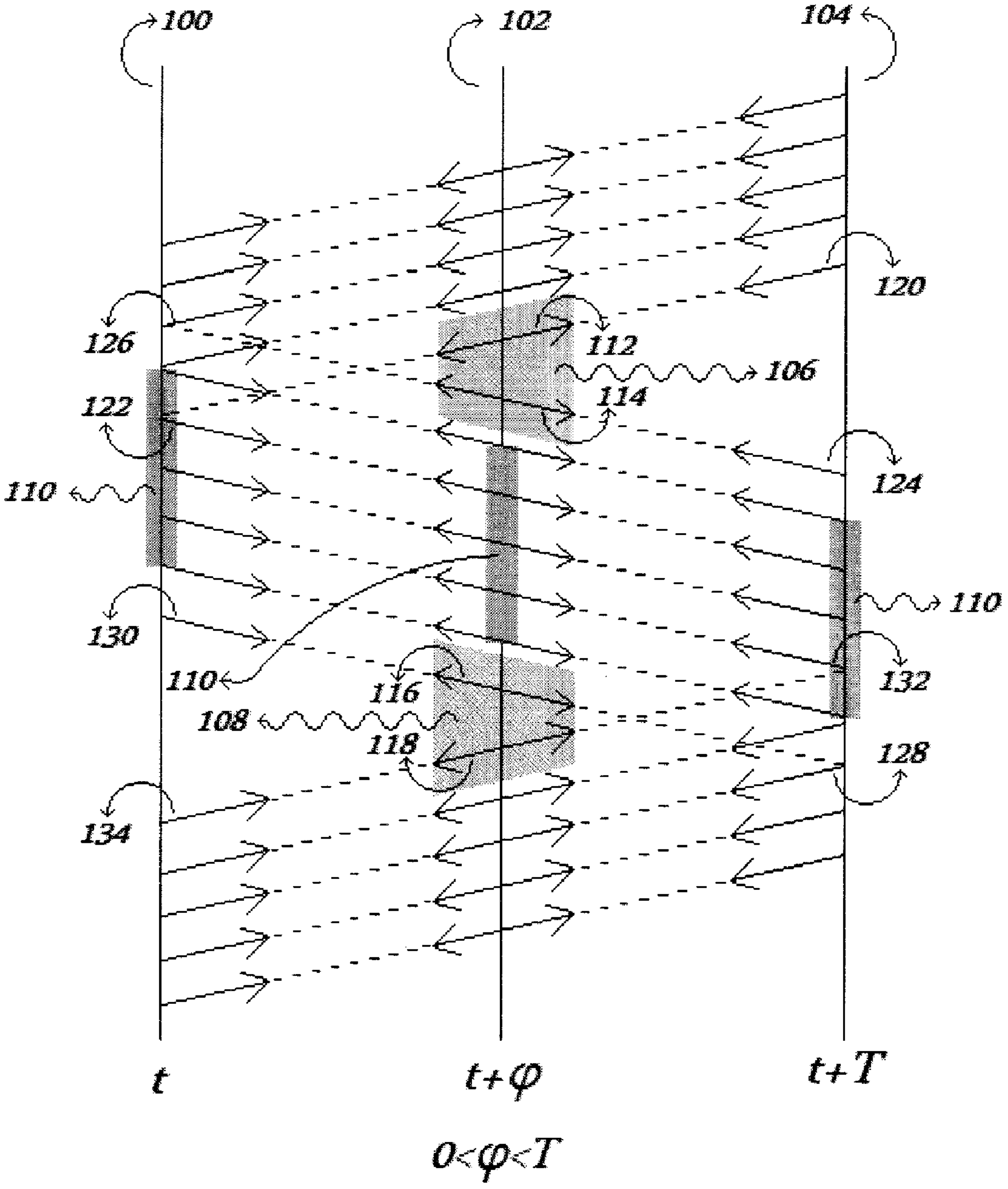

100
102
104
120
126
112
106
114
122
124
110
110
130
110
132
116
108
118
128
134
t
t+φ
t+T
0<φ<T

Figure 1

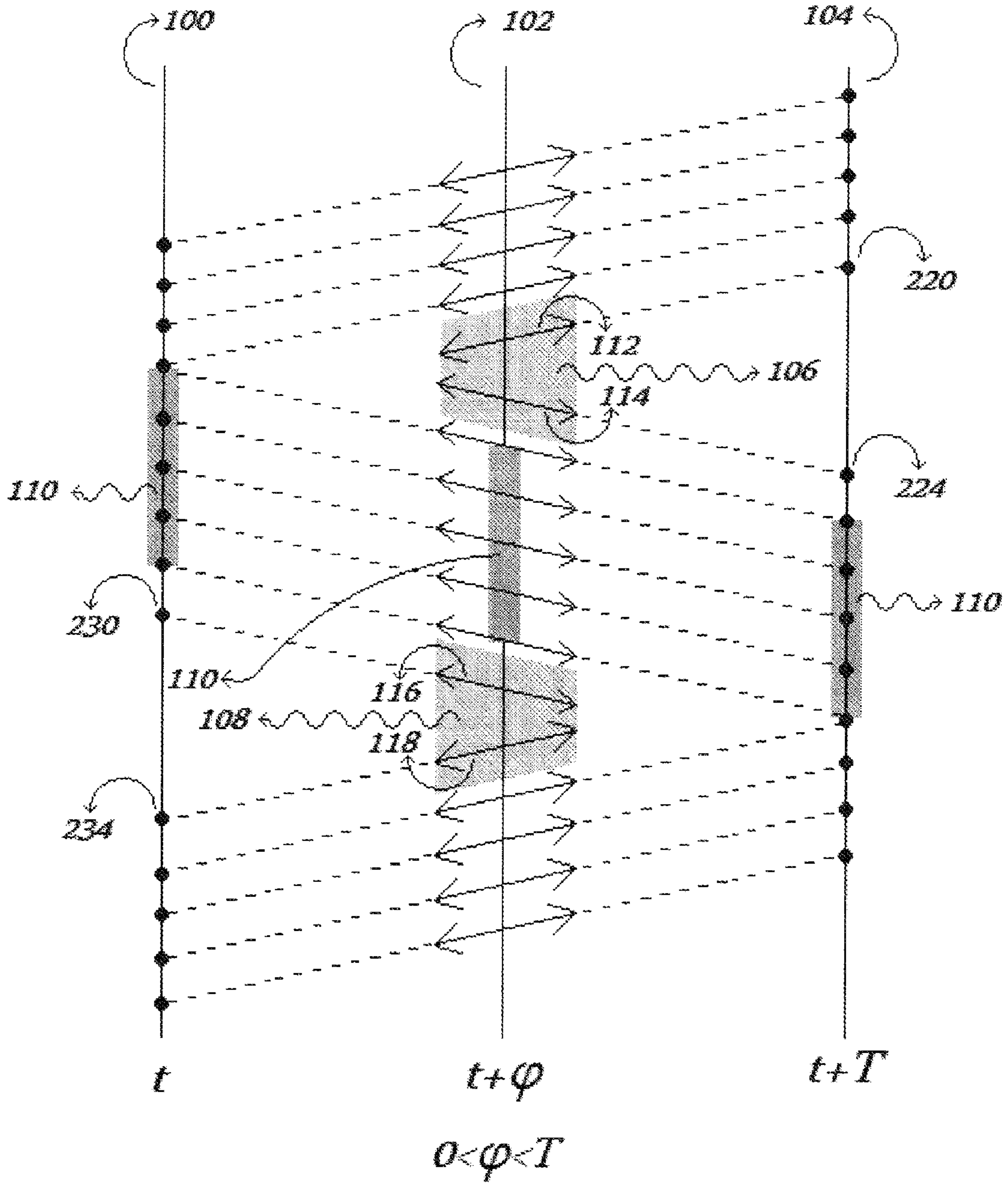

100
102
104
220
112
106
114
224
110
110
230
110
116
108
118
234
t
t+φ
t+T
0<φ<T

Figure 2

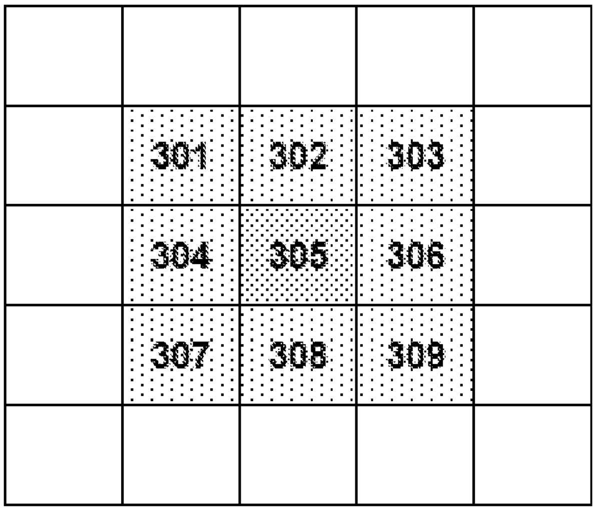


Figure 3

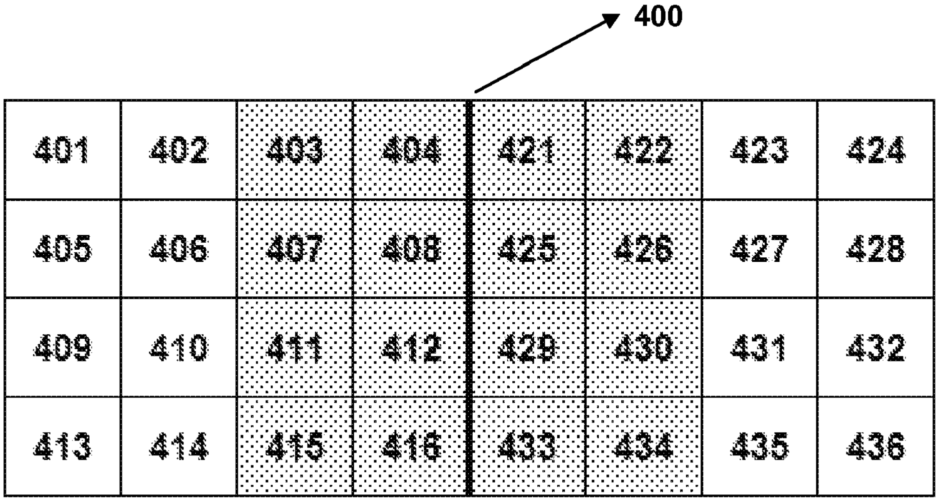
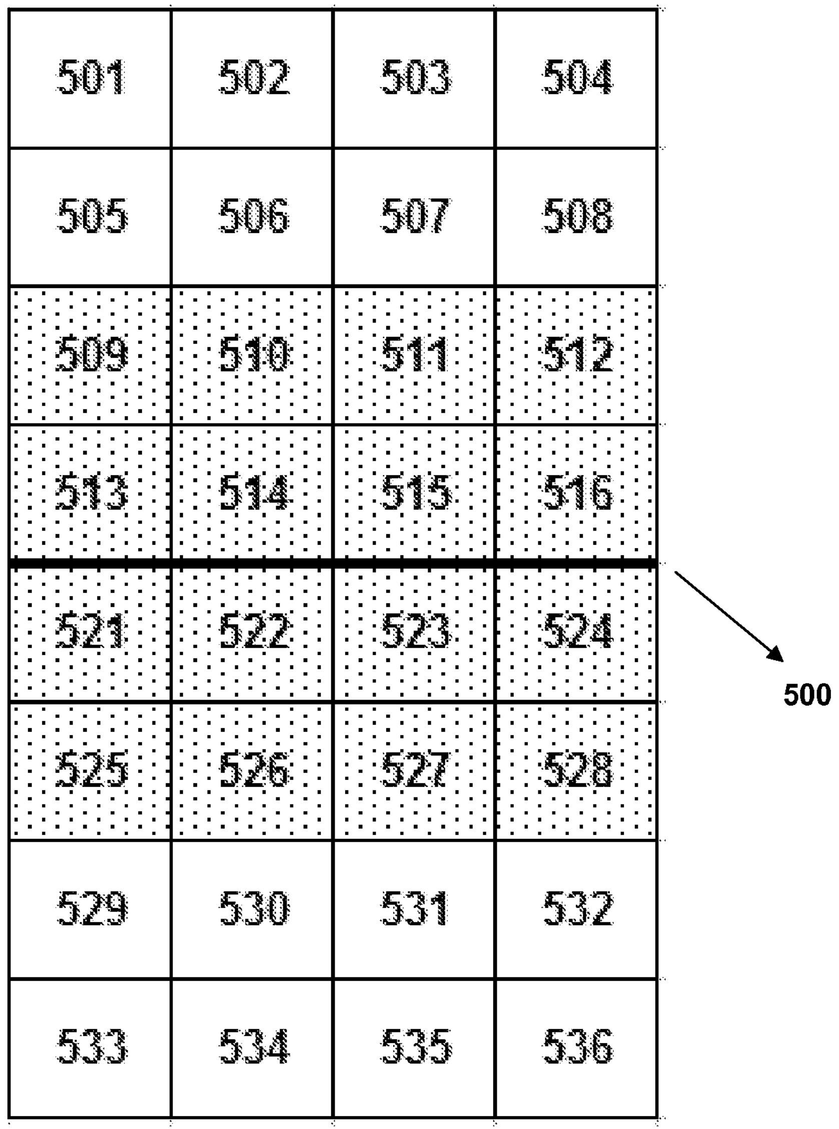
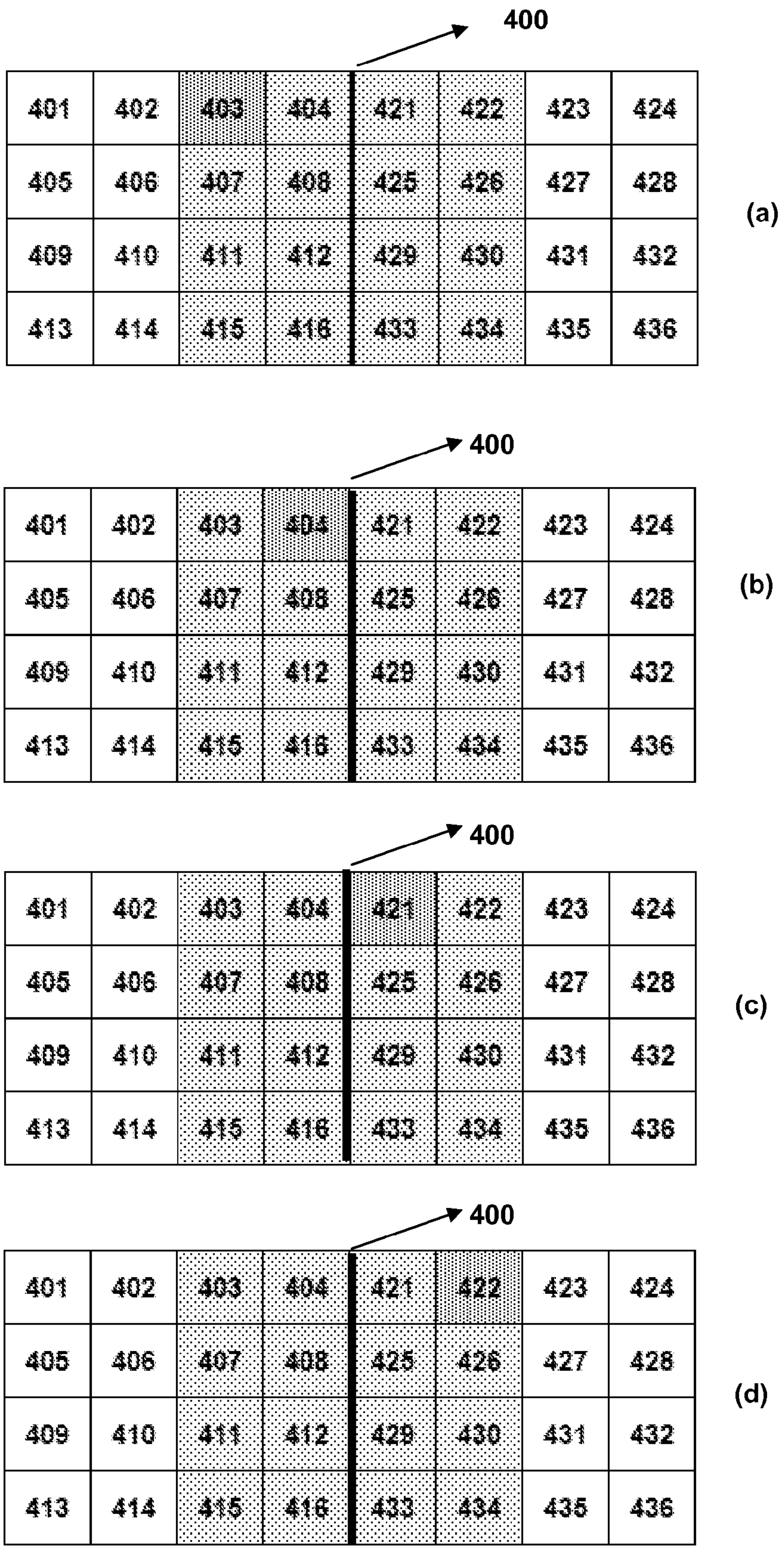


Figure 4



501	502	503	504
505	506	507	508
509	510	511	512
513	514	515	516
521	522	523	524
525	526	527	528
529	530	531	532
533	534	535	536

Figure 5



401	402	403	404	421	422	423	424
405	406	407	408	425	426	427	428
409	410	411	412	429	430	431	432
413	414	415	416	433	434	435	436

(a)

401	402	403	404	421	422	423	424
405	406	407	408	425	426	427	428
409	410	411	412	429	430	431	432
413	414	415	416	433	434	435	436

(b)

401	402	403	404	421	422	423	424
405	406	407	408	425	426	427	428
409	410	411	412	429	430	431	432
413	414	415	416	433	434	435	436

(c)

401	402	403	404	421	422	423	424
405	406	407	408	425	426	427	428
409	410	411	412	429	430	431	432
413	414	415	416	433	434	435	436

(d)

Figure 6

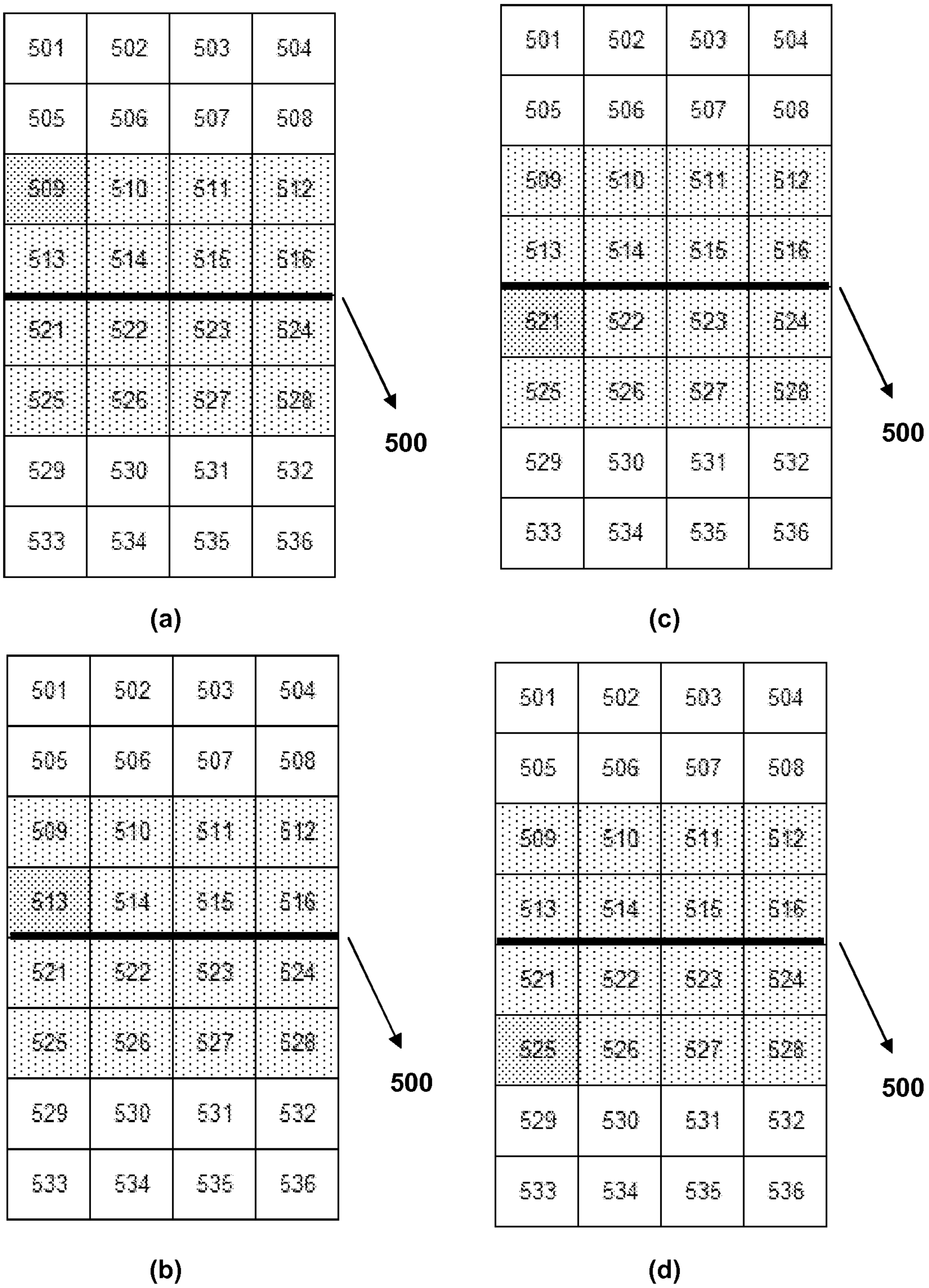

501 502 503 504
505 506 507 508
509 510 511 512
513 514 515 516
521 522 523 524
525 526 527 528
529 530 531 532
533 534 535 536
500
(a)
501 502 503 504
505 506 507 508
509 510 511 512
513 514 515 516
521 522 523 524
525 526 527 528
529 530 531 532
533 534 535 536
500
(c)
501 502 503 504
505 506 507 508
509 510 511 512
513 514 515 516
521 522 523 524
525 526 527 528
529 530 531 532
533 534 535 536
500
(b)
501 502 503 504
505 506 507 508
509 510 511 512
513 514 515 516
521 522 523 524
525 526 527 528
529 530 531 532
533 534 535 536
500
(d)

Figure 7

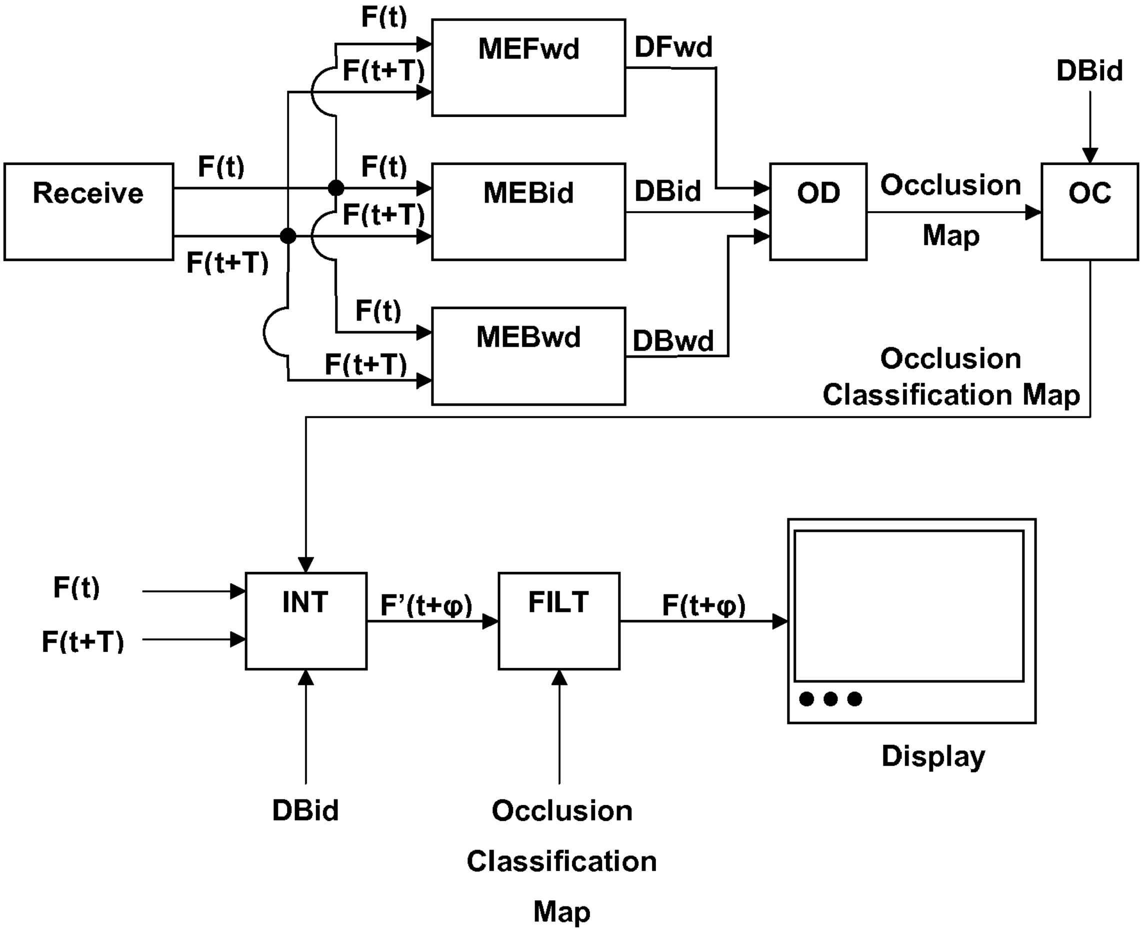


Figure 8

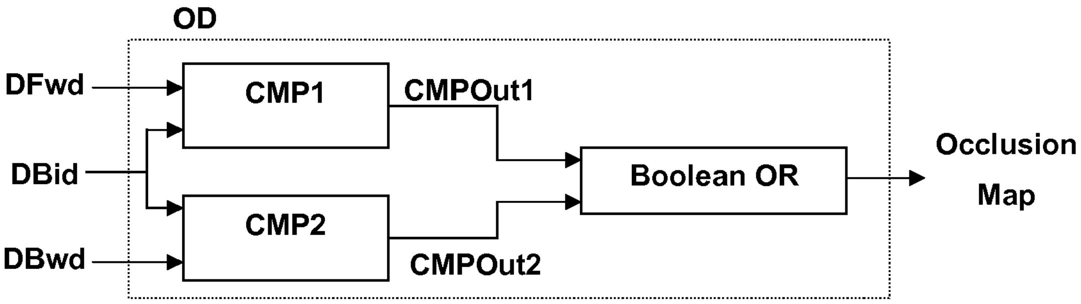


Figure 9

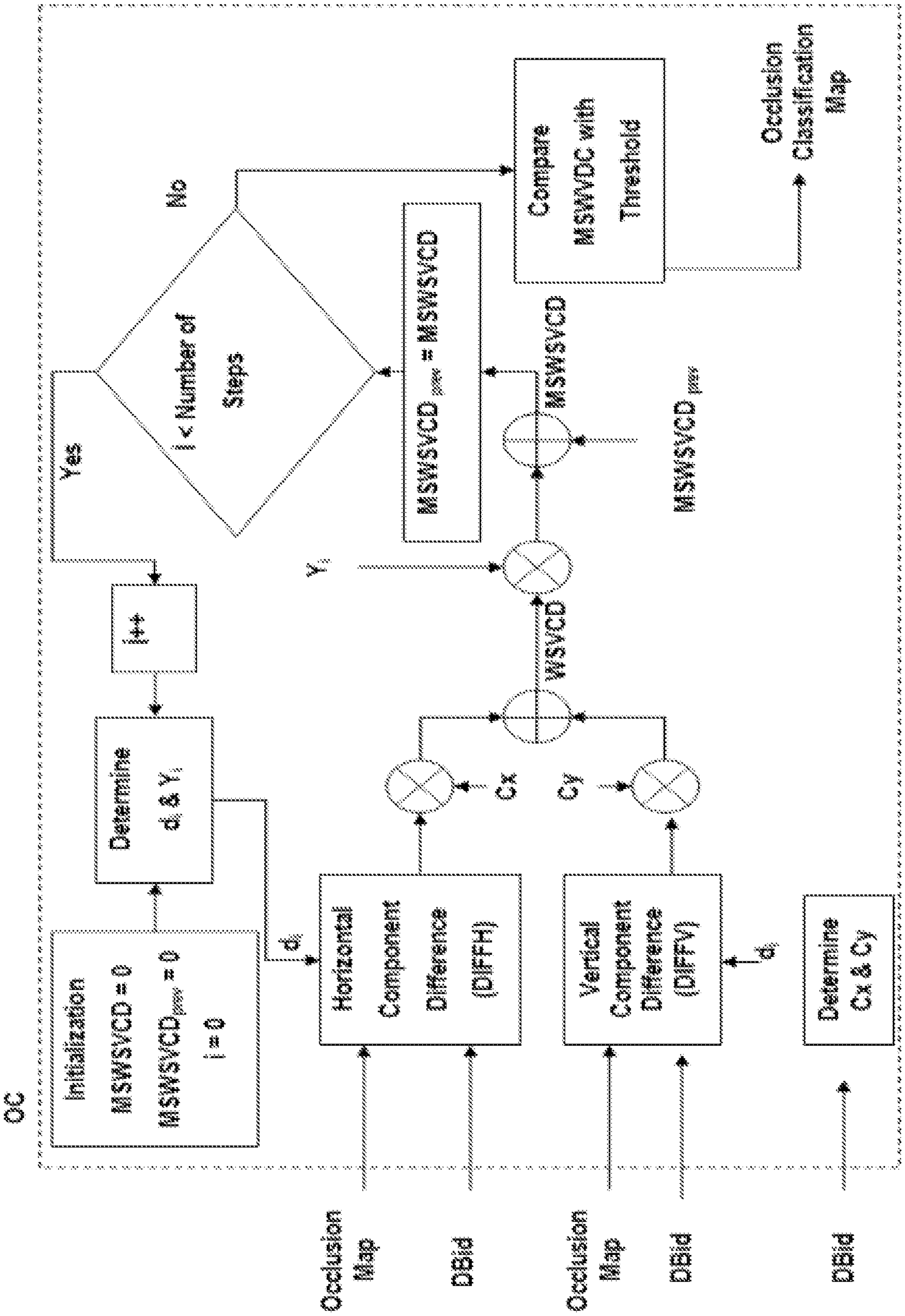

OC
Initialization
MSWSVCD = 0
MSWSVCD_prev = 0
i = 0
Determine
d_i & Y_i
i++
Yes
i < Number of Steps
No
MSWSVCD_prev = MSWSVCD
Compare
MSWVDC with
Threshold
Occlusion
Classification
Map
d_i
Horizontal
Component
Difference
(DIFFH)
Cx
Cy
WSVCD
Y_i
MSWSVCD
MSWSVCD_prev
Vertical
Component
Difference
(DIFFV)
d_i
Determine
Cx & Cy
Occlusion
Map
DBid
Occlusion
Map
DBid
DBid

Figure 10

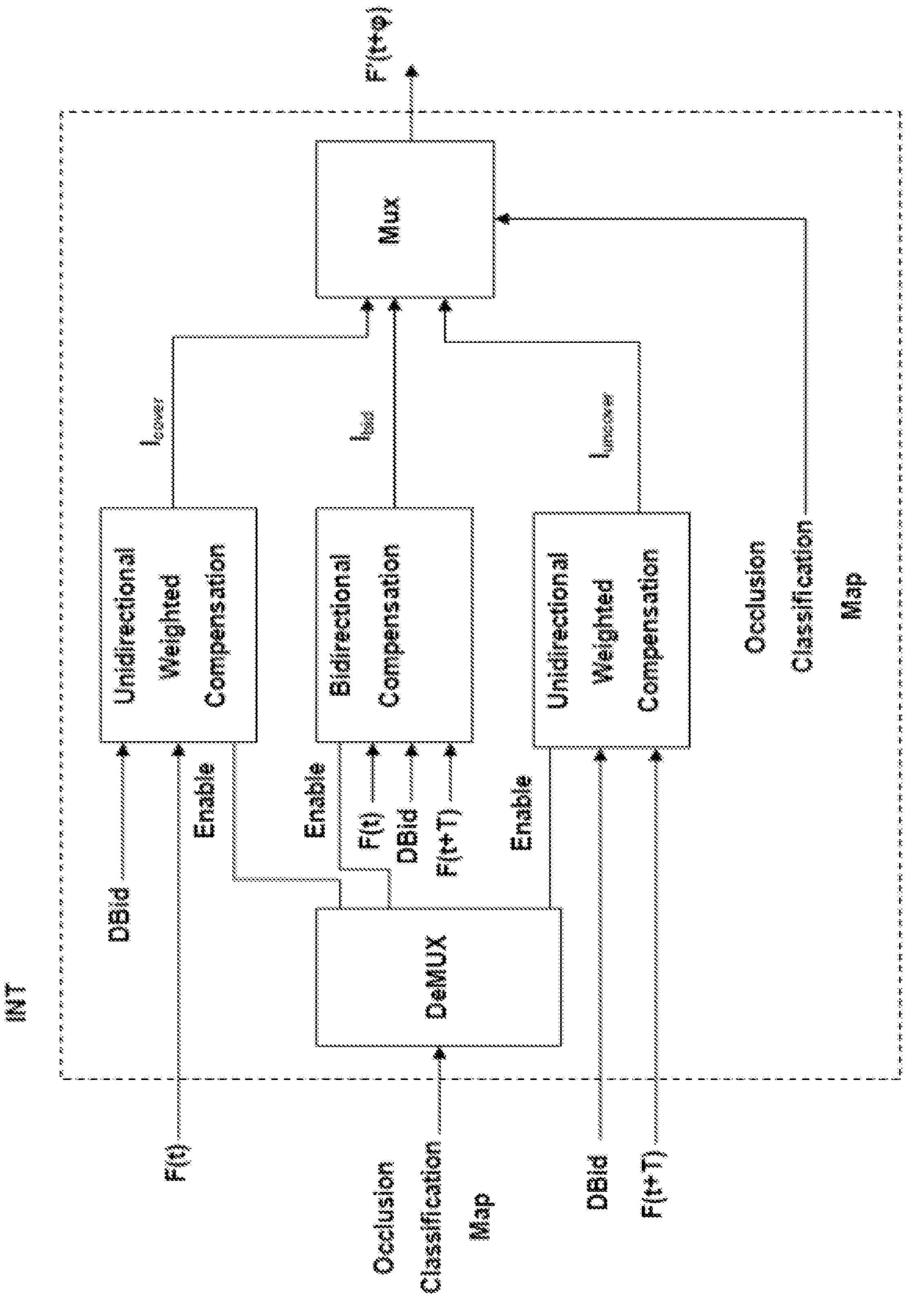

F'(t+φ)
Mux
I_{cover}
I_{bid}
$I_{uncover}$
Unidirectional Weighted Compensation
Bidirectional Compensation
Unidirectional Weighted Compensation
Occlusion Classification Map
DBid
Enable
Enable
F(t)
DBid
F(t+T)
Enable
INT
DeMUX
F(t)
Occlusion Classification Map
DBid
F(t+T)

Figure 11

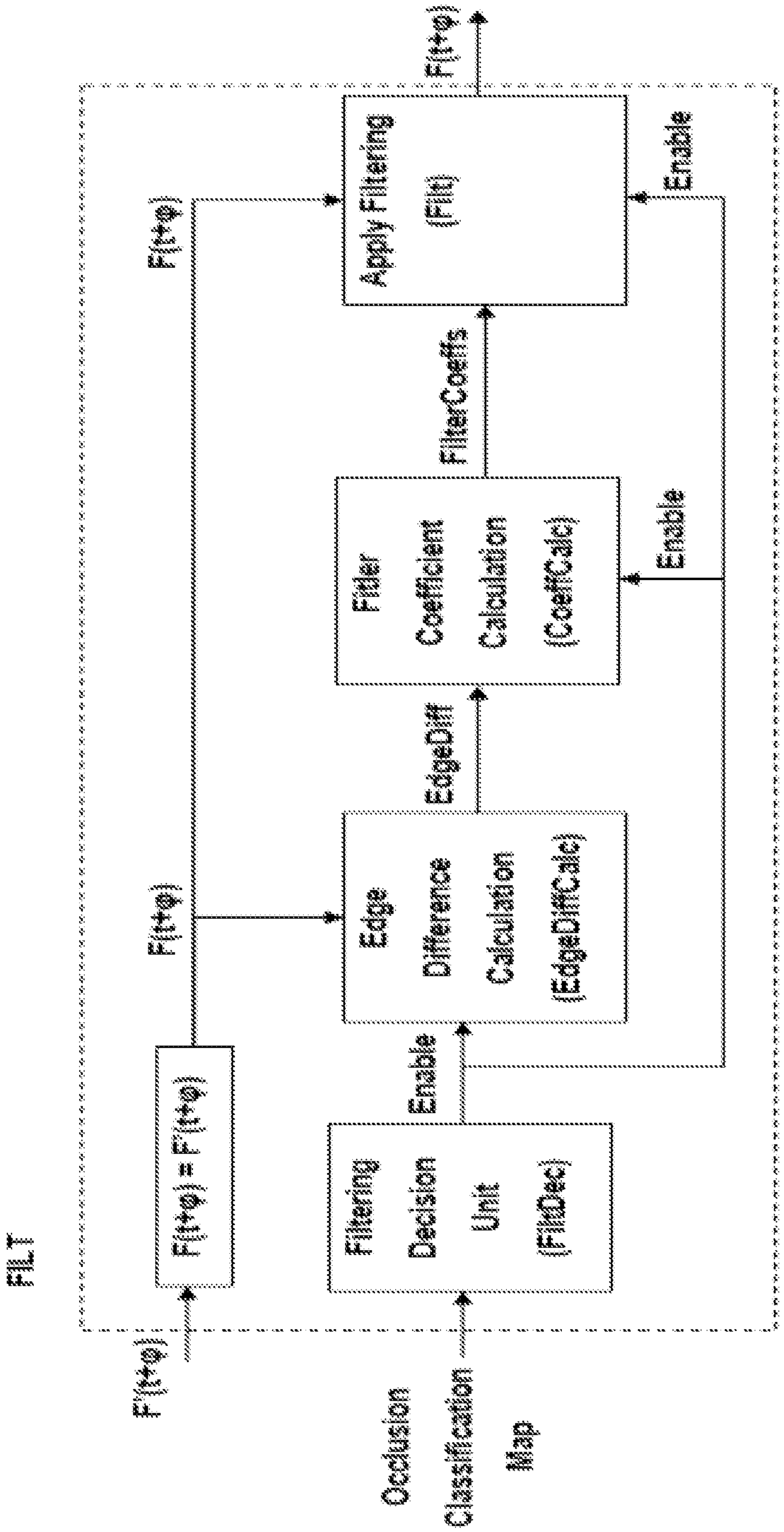

FILT
F'(t+φ)
Occlusion Classification Map
Filtering Decision Unit (FiltDec)
F(t+φ) = F'(t+φ)
Enable
F(t+φ)
Edge Difference Calculation (EdgeDiffCalc)
EdgeDiff
Filter Coefficient Calculation (CoeffCalc)
Enable
FilterCoeffs
F(t+φ)
Apply Filtering (Filt)
Enable
F(t+φ)

Figure 12

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 6219436 B **[0007]**
- US 6487313 B **[0008]**
- WO 9704599 A **[0009]**