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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 August 2010 (05.08.2010)

PCT

(10) International Publication Number
WO 2010/087955 A1

(54) Title: CODING OF DEPTH MAPS

FIG. 3
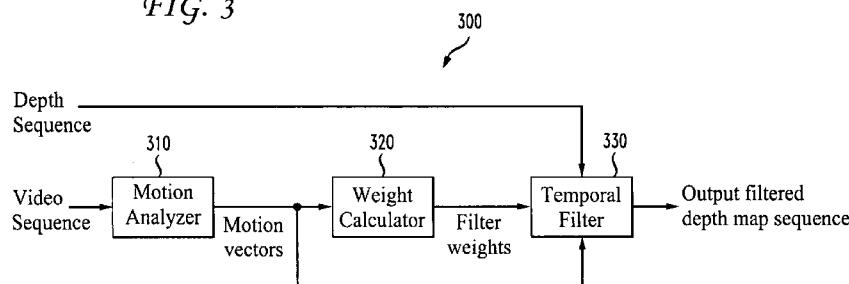
(57) Abstract: Various implementations are described. Several implementations relate to filtering of depth maps. According to a general aspect, a first depth picture is accessed that corresponds to a first video picture. For a given portion of the first depth picture, a co-located video portion of the first video picture is determined. A video motion vector is accessed that indicates motion of the co-located video portion of the first video picture with respect to a second video picture. A second depth picture is accessed that corresponds to the second video picture. A depth portion of the second depth picture is determined, from the given portion of the first depth picture, based on the video motion vector. The given portion of the first depth picture is updated based on the depth portion of the second depth picture.

CODING OF DEPTH MAPS

CROSS-REFERENCE TO RELATED APPLICATIONS

5   This application claims the benefit of U.S. Provisional Application Serial No.
61/206,496, titled "Temporal Filtering of Depth Maps", and filed January 30, 2009, which is
incorporated by reference herein in its entirety for all purposes.

TECHNICAL FIELD

10   Implementations are described that relate to coding systems. Various particular
implementations relate to coding of depth maps

BACKGROUND

          Currently in many application systems, depth maps are obtained by depth estimation
15   methods which find correspondence among different video views. Due to camera setting
mismatches, different shooting positions, and depth estimation errors, the resulting depth
maps may exhibit some artifacts. The use of such depth maps could lead to unacceptable
rendering errors in generated virtual views.

20   SUMMARY

          According to a general aspect, a first depth picture is accessed that corresponds to a
first video picture. For a given portion of the first depth picture, a co-located video portion
of the first video picture is determined. A video motion vector is accessed that indicates
motion of the co-located video portion of the first video picture with respect to a second
25   video picture. A second depth picture is accessed that corresponds to the second video
picture. A depth portion of the second depth picture is determined, from the given portion of
the first depth picture, based on the video motion vector. The given portion of the first depth
picture is updated based on the depth portion of the second depth picture.

          The details of one or more implementations are set forth in the accompanying
30   drawings and the description below. Even if described in one particular manner, it should be
clear that implementations may be configured or embodied in various manners. For
example, an implementation may be performed as a method, or embodied as apparatus, such
as, for example, an apparatus configured to perform a set of operations or an apparatus
storing instructions for performing a set of operations, or embodied in a signal. Other aspects

and features will become apparent from the following detailed description considered in conjunction with the accompanying drawings and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is an example of a left depth map from a left reference view.

Figure 2 is an example of a right depth map from a right reference view.

Figure 3 is a diagram of an implementation of an apparatus for performing temporal filtering of depth maps.

Figure 4 is a diagram of an implementation of a framework for generating nine output views (N = 9) out of 3 input views with depth (K = 3).

Figure 5 is a diagram of an implementation of a video transmission system.

Figure 6 is a diagram of an implementation of a video receiving system.

Figure 7 is a diagram of an implementation of a video processing device.

Figures 8, 9, and 10 are examples of depth maps at three different timestamps around the laptop region in the MPEG test sequence known as "Leaving_Laptop".

Figures 11, 12, and 13 are examples of depth maps around the floor region in the MPEG test sequence known as "Leaving_Laptop".

Figure 14 illustrates an example of a filtering operation applied to a region within the depth map D at time t using its two corresponding regions at time t-1 and time t+1.

Figure 15 is a diagram of an implementation of a method for performing temporal filtering of depth maps.

DETAILED DESCRIPTION

To facilitate new video applications such as three-dimensional television (3DTV) and free-viewpoint video (FVV), a data format that includes multiple video views and the corresponding per-pixel depth maps is commonly used. The data format is referred to as the multi-view plus depth (MVD) format.

In at least one implementation, we propose performing temporal filtering across depth maps at different timestamps to improve the temporal consistency in the depth maps, as well as to remove the false contours in regions with flat depths or smoothly varying depths.

In at least one implementation, we propose to use the video information to identify correspondence among depth maps such that the filtering is applied to corresponding regions/objects across time.

The processed depth maps will typically have improved temporal consistency in static regions and they will have much less banding (false contours). These improvements should not only provide higher rendering quality for the virtual views rendered using the processed depth maps, but also enable higher depth coding efficiency as the depths for static regions
5   become stable (benefits temporal prediction) and the false contours are removed (benefits spatial prediction).

Figure 1 shows an exemplary depth map generated for a sixth view corresponding to an MPEG test sequence known as "Leaving_Laptop", to which the present principles may be applied in accordance with an embodiment of the present principles. Figure 2 shows an
10   exemplary depth map generated for a ninth view corresponding to the MPEG test sequence known as "Leaving_Laptop", to which the present principles may be applied in accordance with an embodiment of the present principles. At the user end, intermediate video views (virtual views) can be generated using, for example, the techniques of depth image based rendering (DIBR), which takes the transmitted or stored video views (reference views) and
15   the corresponding depth maps as input.

Figure 3 shows an exemplary apparatus 300 for performing temporal filtering of depth maps to which the present principles may be applied, in accordance with an embodiment of the present principles. The apparatus 300 includes a motion analyzer 310 having an output connected in signal communication with an input of a weight calculator 320
20   and a third input of a temporal filter 330, for providing motion vectors thereto. An output of the weight calculator 320 is connected in signal communication with a first input of the temporal filter 330, for providing filter weights thereto. An input of the motion analyzer 310 is available as an input of the apparatus 300, for receiving a video sequence. A second input of the temporal filter 330 is available as an input of the apparatus 300, for receiving a depth
25   sequence. An output of the temporal filter 330 is available as an output of the apparatus 300, for outputting a filtered depth map sequence. The operation of the apparatus 300 is described in further detail herein below.

Implementations of Figure 3, as well as other devices described in the application, may receive more than one input signal on a single input terminal or port. For example,
30   implementations one or more of the temporal filter 330 may receive depth sequences, filter weights, and motion vectors on a single input port.

Figure 4 shows an exemplary framework 400 for generating nine output views (N = 9) out of 3 input views with depth (K = 3), to which the present principles may be applied, in accordance with an embodiment of the present principles. The framework 400 involves an

3

auto-stereoscopic 3D display 410, which supports output of multiple views, a first depth image-based renderer 420, a second depth image-based renderer 430, and a buffer for decoded data 440. The decoded data is a representation known as Multiple View plus Depth (MVD) data. The nine views are denoted by V1 through V9. Corresponding depth maps for

5   the three input views are denoted by D1, D5, and D9. Any virtual camera positions in between the captured camera positions (e.g., Pos 1, Pos 2, Pos 3) can be generated using the available depth maps (D1, D5, D9), as shown in Figure 4. As can be seen in Figure 4, the baseline between the actual cameras (V1, V5 and V9) used to capture data can be large. As a result, the correlation between these cameras is significantly reduced and coding efficiency

10   of these cameras may suffer since the coding efficiency would rely on temporal correlation.

Figure 5 shows an exemplary video transmission system 500, to which the present principles may be applied, in accordance with an implementation of the present principles. The video transmission system 500 may be, for example, a head-end or transmission system for transmitting a signal using any of a variety of media, such as, for example, satellite, cable,

15   telephone-line, or terrestrial broadcast. The transmission may be provided over the Internet or some other network.

The video transmission system 500 is capable of generating and delivering compressed video with depth. This is achieved by generating an encoded signal(s) including depth information or information capable of being used to synthesize the depth information

20   at a receiver end that may, for example, have a decoder.

The video transmission system 500 includes an encoder 510 and a transmitter 520 capable of transmitting the encoded signal. The encoder 510 receives video information and generates an encoded signal(s) with depth. The encoder 510 may include sub-modules, including for example an assembly unit for receiving and assembling various pieces of

25   information into a structured format for storage or transmission. The various pieces of information may include, for example, coded or uncoded video, coded or uncoded depth information, and coded or uncoded elements such as, for example, motion vectors, coding mode indicators, and syntax elements.

The transmitter 520 may be, for example, adapted to transmit a program signal having

30   one or more bitstreams representing encoded pictures and/or information related thereto. Typical transmitters perform functions such as, for example, one or more of providing error-correction coding, interleaving the data in the signal, randomizing the energy in the signal, and/or modulating the signal onto one or more carriers. The transmitter may include, or

interface with, an antenna (not shown). Accordingly, implementations of the transmitter 520 may include, or be limited to, a modulator.

Figure 6 shows an exemplary video receiving system 600 to which the present principles may be applied, in accordance with an embodiment of the present principles. The video receiving system 600 may be configured to receive signals over a variety of media, such as, for example, satellite, cable, telephone-line, or terrestrial broadcast. The signals may be received over the Internet or some other network.

The video receiving system 600 may be, for example, a cell-phone, a computer, a set-top box, a television, or other device that receives encoded video and provides, for example, decoded video for display to a user or for storage. Thus, the video receiving system 600 may provide its output to, for example, a screen of a television, a computer monitor, a computer (for storage, processing, or display), or some other storage, processing, or display device.

The video receiving system 600 is capable of receiving and processing video content including video information. The video receiving system 600 includes a receiver 610 capable of receiving an encoded signal, such as for example the signals described in the implementations of this application, and a decoder 620 capable of decoding the received signal.

The receiver 610 may be, for example, adapted to receive a program signal having a plurality of bitstreams representing encoded pictures. Typical receivers perform functions such as, for example, one or more of receiving a modulated and encoded data signal, demodulating the data signal from one or more carriers, de-randomizing the energy in the signal, de-interleaving the data in the signal, and/or error-correction decoding the signal. The receiver 610 may include, or interface with, an antenna (not shown). Implementations of the receiver 610 may include, or be limited to, a demodulator. The decoder 620 outputs video signals including video information and depth information.

Figure 7 shows an exemplary video processing device 700 to which the present principles may be applied, in accordance with an embodiment of the present principles. The video processing device 700 may be, for example, a set top box or other device that receives encoded video and provides, for example, decoded video for display to a user or for storage. Thus, the video processing device 700 may provide its output to a television, computer monitor, or a computer or other processing device.

The video processing device 700 includes a front-end (FE) device 705 and a decoder 710. The front-end device 705 may be, for example, a receiver adapted to receive a program signal having a plurality of bitstreams representing encoded pictures, and to select one or

more bitstreams for decoding from the plurality of bitstreams. Typical receivers perform functions such as, for example, one or more of receiving a modulated and encoded data signal, demodulating the data signal, decoding one or more encodings (for example, channel coding and/or source coding) of the data signal, and/or error-correcting the data signal. The

5   front-end device 705 may receive the program signal from, for example, an antenna (not shown). The front-end device 705 provides a received data signal to the decoder 710.

The decoder 710 receives a data signal 720. The data signal 720 may include, for example, one or more Advanced Video Coding (AVC), Scalable Video Coding (SVC), or Multi-view Video Coding (MVC) compatible streams.

10   AVC refers more specifically to the existing International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) Moving Picture Experts Group-4 (MPEG-4) Part 10 Advanced Video Coding (AVC) standard/International Telecommunication Union, Telecommunication Sector (ITU-T) H.264 Recommendation (hereinafter the "H.264/MPEG-4 AVC Standard" or variations thereof, such as the "AVC

15   standard" or simply "AVC").

MVC refers more specifically to a multi-view video coding ("MVC") extension (Annex H) of the AVC standard, referred to as H.264/MPEG-4 AVC, MVC extension (the "MVC extension" or simply "MVC").

SVC refers more specifically to a scalable video coding ("SVC") extension (Annex

20   G) of the AVC standard, referred to as H.264/MPEG-4 AVC, SVC extension (the "SVC extension" or simply "SVC").

The decoder 710 decodes all or part of the received signal 720 and provides as output a decoded video signal 730. The decoded video 730 is provided to a selector 750. The device 700 also includes a user interface 760 that receives a user input 770. The user

25   interface 760 provides a picture selection signal 780, based on the user input 770, to the selector 750. The picture selection signal 780 and the user input 770 indicate which of multiple pictures, sequences, scalable versions, views, or other selections of the available decoded data a user desires to have displayed. The selector 750 provides the selected picture(s) as an output 790. The selector 750 uses the picture selection information 780 to

30   select which of the pictures in the decoded video 730 to provide as the output 790.

In various implementations, the selector 750 includes the user interface 760, and in other implementations no user interface 760 is needed because the selector 750 receives the user input 770 directly without a separate interface function being performed. The selector 750 may be implemented in software or as an integrated circuit, for example. In one

implementation, the selector 750 is incorporated with the decoder 710, and in another implementation, the decoder 710, the selector 750, and the user interface 760 are all integrated.

5   In one application, front-end 705 receives a broadcast of various television shows and selects one for processing. The selection of one show is based on user input of a desired channel to watch. Although the user input to front-end device 705 is not shown in Figure 7, front-end device 705 receives the user input 770. The front-end 705 receives the broadcast and processes the desired show by demodulating the relevant part of the broadcast spectrum, and decoding any outer encoding of the demodulated show. The front-end 705 provides the

10   decoded show to the decoder 710. The decoder 710 is an integrated unit that includes devices 760 and 750. The decoder 710 thus receives the user input, which is a user-supplied indication of a desired view to watch in the show. The decoder 710 decodes the selected view, as well as any required reference pictures from other views, and provides the decoded view 790 for display on a television (not shown).

15   Continuing the above application, the user may desire to switch the view that is displayed and may then provide a new input to the decoder 710. After receiving a "view change" from the user, the decoder 710 decodes both the old view and the new view, as well as any views that are in between the old view and the new view. That is, the decoder 710 decodes any views that are taken from cameras that are physically located in between the

20   camera taking the old view and the camera taking the new view. The front-end device 705 also receives the information identifying the old view, the new view, and the views in between. Such information may be provided, for example, by a controller (not shown in Figure 7) having information about the locations of the views, or the decoder 710. Other implementations may use a front-end device that has a controller integrated with the front-

25   end device.

The decoder 710 provides all of these decoded views as output 790. A post-processor (not shown in Figure 7) interpolates between the views to provide a smooth transition from the old view to the new view, and displays this transition to the user. After transitioning to the new view, the post-processor informs (through one or more communication links not

30   shown) the decoder 710 and the front-end device 705 that only the new view is needed. Thereafter, the decoder 710 only provides as output 790 the new view.

The system 700 may be used to receive multiple views of a sequence of images, and to present a single view for display, and to switch between the various views in a smooth manner. The smooth manner may involve interpolating between views to move to another

7

view. Additionally, the system 700 may allow a user to rotate an object or scene, or otherwise to see a three-dimensional representation of an object or a scene. The rotation of the object, for example, may correspond to moving from view to view, and interpolating between the views to obtain a smooth transition between the views or simply to obtain a three-dimensional representation. That is, the user may "select" an interpolated view as the "view" that is to be displayed.

For 3DTV and FVV applications, virtual views can be rendered using DIBR techniques which typically take multiple video views and the corresponding depth maps (such as MVD format) as input. Typically, the depth maps are obtained by depth estimation routines which establish pixel correspondence among video views by minimizing a certain matching cost. Since the depth values are estimated instead of being measured by active devices (such as range cameras), the depth maps are very likely to exhibit estimation errors, leading to poor rendering quality in the generated virtual views.

One possible artifact in the estimated depth maps is the temporal inconsistency in static regions. Figures 8, 9, and 10 show respective exemplary depth maps at three different timestamps around the laptop region in the MPEG test sequence known as "Leaving_Laptop". In particular, Figure 8 shows an exemplary depth map 800 for frame 5, Figure 9 shows an exemplary depth map 900 for frame 10, and Figure 10 shows an exemplary depth map 1000 for frame 15 of the MPEG test sequence "Leaving_Laptop". As the depth estimation is performed on a frame-by-frame basis, no mechanism was developed to check temporal consistency (we refer to this as a "temporally independent depth map"). It can be observed that the laptop boundary and also its depth value change from frame to frame, which will typically lead to flickering artifacts in the rendered virtual views. Furthermore, encoding depth maps with such a temporal inconsistency problem would result in a higher bitrate, as the depth values are changing across time.

Another often observed depth map artifact is banding (false contours) in regions with smoothly varying depth values. Figures 11, 12, and 13 show respective exemplary depth maps around the floor region in the MPEG test sequence known as "Leaving_Laptop". In particular, Figure 11 shows an exemplary depth map 1100 for frame 5; Figure 12 shows an exemplary depth map 1200 for frame 10; and Figure 13 shows an exemplary depth map 1300 for frame 15 of the MPEG test sequence "Leaving_Laptop". From Figures 11, 12, and 13, we see that the depth maps on the floor region exhibits some artificial contours. The depth values in such region should be changing smoothly. These contours could reduce the coding

efficiency on depth maps because such patterns are typically harder to encode than the supposed smooth variation.

To address these problems, at least one implementation described in this disclosure proposes temporal filtering techniques for the estimated depth maps to improve the temporal
5   consistency as well as to reduce or remove the false contours. The video information is used to identify corresponding regions for temporal filtering to be applied. The resulting depth maps, with improved temporal consistency and less banding (false contours), will provide higher rendering quality for the virtual views, and enable higher depth coding efficiency.

To improve temporal consistency and to remove artifacts in estimated depth maps, in
10   at least one implementation, it is proposed to perform temporal filtering after an initial depth estimation has been performed (for example, after temporally independent depth maps are estimated). In this scenario, to filter (or, more generally, update) a depth map at a given timestamp, depth maps from both previous and future timestamps can be used. The temporal window of the depth maps that are to be involved in the filtering process can be determined
15   by locating corresponding regions available in the depth maps at different timestamps. To identify corresponding regions at different timestamps, the motion information in the video view (instead of information in the depth) is used. Since the goal is to remove erroneous depth estimation results, such as changing boundaries of stationary regions and artificial contours, a filtering operation(s) with low-pass characteristics is utilized in at least one
20   implementation. For example, temporal weighted averaging and median filtering are some exemplary candidate processes. In the following, we will provide one possible embodiment of the ideas proposed that relate to temporal filtering of depth maps with motion information from video. A generic block diagram of the filtering process according to one embodiment of the present principles is provided in Fig. 3.
25

Embodiment example

For a given view, let us denote $I(x,y,t)$ and $D(x,y,t)$ as the video luminance pixel value and the corresponding estimated depth map value at pixel location (x,y) and time t, respectively. We presume that we want to perform temporal filtering for a region in D at
30   time t: { $D(x,y,t) \mid x_0 \leq x \leq x_1, y_0 \leq y \leq y_1$ }. The following steps describe how the temporal filtering is achieved.

1.   Identify corresponding regions at different timestamps based on motion information
       from video:

In an embodiment, the identification of the corresponding regions at different timestamps based on motion information from video is performed using the motion analyzer 310 of FIG. 3.

5 Since it is known that the depth information could be erroneous, the co-located region in video frame { $I(x,y,t) \mid x_0 \leq x \leq x_1, y_0 \leq y \leq y_1$ } is used to extract motion information and establish correspondence across different times. Any conventional motion analysis techniques can be utilized for this purpose to identify corresponding regions between $I(x,y,t')$ and $I(x,y,t)$, where $t' \neq t$. For example, a simple block-based motion search as typically performed in video coding is one possible choice. A temporal boundary can be set up such 10 that the filtering is performed within a reasonable temporal distance (for example $t-T \leq t' \leq t+T$). Due to moving objects, it is possible that a given region may not have a correspondence at some timestamps. This can possibly be detected by setting some threshold in the motion search cost function.

After this motion analysis stage, for the given video region { $I(x,y,t) \mid x_0 \leq x \leq x_1$, 15 $y_0 \leq y \leq y_1$ }, we obtain motion vectors ($mvx_{t'}, mvy_{t'}$) for the corresponding region between t and t', to identify candidate correspondences available at different timestamps t': { $I(x,y,t') \mid x_0 \leq x-mvx_{t'} \leq x_1, y_0 \leq y-mvy_{t'} \leq y_1, t-T \leq t' \leq t+T$ }. Note that the depth correspondence for the given region { $D(x,y,t) \mid x_0 \leq x \leq x_1, y_0 \leq y \leq y_1$ } can be established using these motion vectors to obtain corresponding regions { $D(x,y,t') \mid x_0 \leq x-mvx_{t'} \leq x_1, y_0 \leq y-mvy_{t'} \leq y_1, t-T \leq t' \leq t+T$ }.

20 This operation will be performed in Step 3(temporal averaging) described below.

2.   Determine filter weights:

In an embodiment, the determination of the filter weights is performed using the weight calculator 320 of FIG. 3.

25 To perform filtering, each depth correspondence identified by copying the motion vectors found in Step 1 is given a weight $w_{t'}$. The filtered depth, within ($x_0 \leq x \leq x_1, y_0 \leq y \leq y_1$) is $D''(x,y,t) = \sum w_{t'} \times D(x+mvx_{t'}, y+mvy_{t'}, t')$, where $\sum w_{t'} = 1$, and both summations are performed over t' falling within the selected time range of T. The simplest averaging method is to use uniform weighting where the weights w are the same regardless of the time t'. In at 30 least one implementation, we also propose other properties to be considered when determining weights.

Motion criterion: For objects that remain static across multiple timestamps, their depth should not change. Thus, if it is observed that ($mvx_{t'}, mvy_{t'}$) are mostly zero, we have better

confidence that applying temporal filtering can improve temporal consistency for this region in the depth map. On the other hand, a moving object may also change its depth across time. An insensitive temporal filter may introduce some error to the depth map. Based on this rationale, we can apply a larger weight w to the correspondence with smaller motion, and

5   apply a smaller weight (or even $w = 0$) to the correspondence with larger motion.

Temporal distance criterion: Beside the motion criterion above, another thing worth considering is the temporal distance between t' and t. For a typical low-pass filtering scenario, the largest weight is commonly assigned to $t' = t$ and the weights decrease as t'

10   moves away from t. For example, weights $0.5 \times \{1, -2, 4, -2, 1\}$ can be assigned for $t' = \{t-2, t-1, t, t+1, t+2\}$. Note that if there are some timestamps at which no correspondence were found in Step 1, then the weights have to be adjusted accordingly to ensure $\sum w_{t'} = 1$. For example, if no match exists at $t' = t + 1$, then the weights will become $0.25 \times \{1, -2, 4, 1\}$ for $t' = \{t-2, t-1, t, t+2\}$. Note that the use of negative weights allows, for example, the filter to

15   have a frequency response approaching that of an ideal low-pass filter.

Special case (Median filter):

If the selected filter is a median filter, then for each pixel within $x_0 \le x \le x_1$, $y_0 \le y \le y_1$, the filtered value $D''(x,y,t)$ is the median of the pixels in the corresponding location(s) identified

20   in Step 1. Note that in this case the filtering weights are assigned on a pixel basis since the temporal median of each pixel may not come from the same timestamp. Also note that a median filter can be described as a weighted filter with a weight of one on the median value.

3.   <u>Temporal averaging:</u>

25   In an embodiment, the temporal averaging is performed using the temporal filter 330 of FIG. 3.

To perform filtering for the given region $\{ D(x,y,t) \mid x_0 \le x \le x_1, y_0 \le y \le y_1 \}$, the corresponding regions $\{ D(x,y,t') \mid x_0 \le x - mvx_{t'} \le x_1, y_0 \le y - mvy_{t'} \le y_1, t-T \le t' \le t+T \}$ are obtained using the motion vectors $(mvx_{t'}, mvy_{t'})$ found in Step 1. Now as the weights $w_{t'}$ are

30   determined, the filtered depth values D'' can be calculated as follows, with the summation being made over the various values of t' that are selected within the timestamp range T:

$$D''(x,y,t) = \sum w_{t'} \times D(x+mvx_{t'}, y+mvy_{t'}, t'), \text{ where } \sum w_{t'} = 1$$

The filtered depth values will replace the initial depth values in order to generate the filtered depth maps.

After filtering is done for the entire depth map $D(x,y,t)$ at time t, the procedure will be repeated for the next depth map at $t+1$. Note that at this point, one can use the already
5   filtered map $D''(x,y,t)$ to perform filtering for $D(x,y,t+1)$, or instead use the original $D(x,y,t)$. Both methods can be considered for a general purpose. To avoid the potential problem of over filtering, one implementation uses the initial depth maps for filtering the depth map at $t+1$. In contrast, another implementation uses the already filtered map in an effort to provide greater continuity between earlier depth maps and the current depth map.

10   Figure 14 illustrates an example of a filtering operation applied to a region within the depth map D at time t using its two corresponding regions at time $t-1$ and time $t+1$, and also depicts the relationship between different regions, in accordance with an embodiment of the present principles. For the given depth region, its co-located region within the video frame I at time t is first obtained. Then, motion analysis is performed to identify correspondence
15   between this co-located video region and regions in other video frames, with motion vectors (mvx_{t-1}, mvy_{t-1}) and (mvx_{t+1}, mvy_{t+1}) as output. The corresponding regions in depth maps at time $t-1$ and time $t+1$, depicted with diagonal stripes, are obtained using these motion vectors. Finally, the filter weights will be applied to the corresponding depth regions (diagonal stripes), and possibly to the given depth region at time t, to compute the filtered
20   value for the given depth region at time t.

Figure 15 shows an exemplary method for performing temporal filtering of depth maps, in accordance with an embodiment of the present principles. At step 1510, the depth map at time t and the corresponding video are obtained as inputs. At step 1515, a region in the depth map to be filtered is selected. At step 1520, motion analysis for the co-located
25   region in the video frame is performed, and the resultant motion information is recorded. At step 1525, filter weights are determined (e.g., based on motion, temporal criteria, and so forth). At step 1530, using the motion information, the corresponding regions in the other depth maps are obtained for the depth region to be filtered. At step 1535, the region is filtered along with its corresponding regions in the other depth maps and using the
30   determined weights. At step 1540, it is determined whether or not the current depth map is done. If so, then control is passed to a step 1545. Otherwise, control is returned to the step 1515. At step 1545, it is determined whether or not there are more depth maps at $t = t + 1$ to filter. If so, then control is returned to the step 1510. Otherwise, the method is terminated.

As noted earlier, the operation of filtering may be referred to as updating. Regardless of the term used, the operation of filtering/updating a given portion of a depth map may involve accessing and using depth values from other depth maps. Indeed, in various implementations, filtering/updating a given portion of a given depth map only involves

5   accessing and using depth values from other depth maps because the given depth map is not considered.

Various implementations update a given portion of a depth map by using depth maps from other views at the same timestamp, and/or depth maps from other views at different timestamps. The depth maps from other views may be used in addition to, or in lieu of,

10   depth maps from the same view at different timestamps. Certain implementations that use depth maps from the same view are able to address, for example, the problem of flickering. Similarly, the use of depth maps from different views may be particularly suited to addressing one or more specific artifacts or types of noise.

Various implementations filter/update measured depth maps, in which the depth

15   values have been measured rather than estimated. Measured depth maps may exhibit different artifacts from estimated depth maps. However, the present principles may provide the same or additional advantages, particularly if different equipment is used to measure different depth maps. The equipment may be different, for example, if different views are considered, or if equipment was changed for a given view, or if a given view has multiple

20   measuring devices that are used under different conditions.

We have described certain implementations that determine a motion vector for a video block, and use that same motion vector to find a correspondence between depth maps to perform a filtering operation on a depth block. However, other implementations use a variation of the motion vector instead of, in or in addition to, the exact motion vector, to find

25   correspondence between depth maps to perform the filtering operation. For example, in one implementation, the exact motion vector is used to find a correspondence between depth maps, but then that motion vector is modified so that a different correspondence is determined. Then both of the correspondences are used in the filtering operation. Such an implementation provides a larger correspondence and, therefore, provides more candidates

30   for the filtering operation. One such modification of a motion vector is to vary the motion vector slightly so that the correspondence points to a different area that is still in the same object or area (for example, foreground or background).

In this application the term motion vector is typically applied to a given block at a time t to indicate the motion of that block with respect to a picture at a different time (for

13

example, having a different timestamp). The motion vector, however, is understood to characterize the relative motion between the two pictures. Therefore, the motion vector applies to the corresponding blocks in both of the pictures.

Several of the implementations and features described in this application may be used
5   in the context of the H.264/MPEG-4 AVC (AVC) Standard, or the AVC standard with the MVC extension, or the AVC standard with the SVC extension. However, these implementations and features may be used in the context of another standard (existing or future), or in a context that does not involve a standard. We thus provide one or more implementations having particular features and aspects. However, features and aspects of
10 described implementations may also be adapted for other implementations.

Additionally, many implementations may be implemented in one or more of an encoder, a decoder, a processor (e.g., a post-processor processing output from a decoder, a pre-processor providing input to an encoder, etc.), and so forth. In at least one implementation, updating the depth values may be performed before encoding depth. In at
15 least one other implementation, depth is not calculated at the encoder, but rather is calculated a receiver/television or at the video source without encoding and transmitting the depth. Instead, only the video may be transmitted with no depth images. Then, for example, the receiver can estimate depth because the receiver wants to be able to render new views for a television that is connected to the receiver. Further, other implementations are contemplated
20 by this disclosure.

We thus provide one or more implementations having particular features and aspects. However, features and aspects of described implementations may also be adapted for other implementations. Implementations may signal information using a variety of techniques including, but not limited to, SEI messages, slice headers, other high level syntax, non-high-
25 level syntax, out-of-band information, datastream data, and implicit signaling. Accordingly, although implementations described herein may be described in a particular context, such descriptions should in no way be taken as limiting the features and concepts to such implementations or contexts.

Reference in the specification to "one embodiment" or "an embodiment" or "one
30 implementation" or "an implementation" of the present principles, as well as other variations thereof, mean that a particular feature, structure, characteristic, and so forth described in connection with the embodiment is included in at least one embodiment of the present principles. Thus, the appearances of the phrase "in one embodiment" or "in an embodiment" or "in one implementation" or "in an implementation", as well any other variations,

appearing in various places throughout the specification are not necessarily all referring to the same embodiment.

It is to be appreciated that the use of any of the following "/", "and/or", and "at least one of", for example, in the cases of "A/B", "A and/or B" and "at least one of A and B", is

5   intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of both options (A and B). As a further example, in the cases of "A, B, and/or C" and "at least one of A, B, and C" and "at least one of A, B, or C", such phrasing is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of the third

10   listed option (C) only, or the selection of the first and the second listed options (A and B) only, or the selection of the first and third listed options (A and C) only, or the selection of the second and third listed options (B and C) only, or the selection of all three options (A and B and C). This may be extended, as readily apparent by one of ordinary skill in this and related arts, for as many items listed.

15   Also, as used herein, the words "picture" and "image" are used interchangeably and refer, for example, to all or part of a still image or all or part of a picture from a video sequence. As is known, a picture may be a frame or a field. Additionally, as used herein, a picture may also be a subset of a frame such as, for example, a top half of a frame or a single macroblock. As another example, a depth picture may be, for example, a complete depth

20   map or a partial depth map that only includes depth information for, for example, a single macroblock of a corresponding video frame.

The implementations described herein may be implemented in, for example, a method or a process, an apparatus, a software program, a data stream, or a signal. Even if only discussed in the context of a single form of implementation (for example, discussed only as a

25   method), the implementation of features discussed may also be implemented in other forms (for example, an apparatus or program). An apparatus may be implemented in, for example, appropriate hardware, software, and firmware. The methods may be implemented in, for example, an apparatus such as, for example, a processor, which refers to processing devices in general, including, for example, a computer, a microprocessor, an integrated circuit, or a

30   programmable logic device. Processors also include communication devices, such as, for example, computers, cell phones, portable/personal digital assistants ("PDAs"), and other devices that facilitate communication of information between end-users.

Implementations of the various processes and features described herein may be embodied in a variety of different equipment or applications, particularly, for example,

equipment or applications associated with data encoding and decoding. Examples of such equipment include an encoder, a decoder, a post-processor processing output from a decoder, a pre-processor providing input to an encoder, a video coder, a video decoder, a video codec, a web server, a set-top box, a laptop, a personal computer, a cell phone, a PDA, and other

5   communication devices. As should be clear, the equipment may be mobile and even installed in a mobile vehicle.

Additionally, the methods may be implemented by instructions being performed by a processor, and such instructions (and/or data values produced by an implementation) may be stored on a processor-readable medium such as, for example, an integrated circuit, a software

10   carrier or other storage device such as, for example, a hard disk, a compact diskette, a random access memory ("RAM"), or a read-only memory ("ROM"). The instructions may form an application program tangibly embodied on a processor-readable medium. Instructions may be, for example, in hardware, firmware, software, or a combination. Instructions may be found in, for example, an operating system, a separate application, or a

15   combination of the two. A processor may be characterized, therefore, as, for example, both a device configured to carry out a process and a device that includes a processor-readable medium (such as a storage device) having instructions for carrying out a process. Further, a processor-readable medium may store, in addition to or in lieu of instructions, data values produced by an implementation.

20   As will be evident to one of skill in the art, implementations may produce a variety of signals formatted to carry information that may be, for example, stored or transmitted. The information may include, for example, instructions for performing a method, or data produced by one of the described implementations, such as syntax or a depth map or parameters for processing one or more depth maps. For example, a signal may be formatted

25   to carry as data the rules for writing or reading the syntax of a described embodiment, or to carry as data the actual syntax-values written by a described embodiment. Such a signal may be formatted, for example, as an electromagnetic wave (for example, using a radio frequency portion of spectrum) or as a baseband signal. The formatting may include, for example, encoding a data stream and modulating a carrier with the encoded data stream. The

30   information that the signal carries may be, for example, analog or digital information. The signal may be transmitted over a variety of different wired or wireless links, as is known. The signal may be stored on a processor-readable medium.

A number of implementations have been described. Nevertheless, it will be understood that various modifications may be made. For example, elements of different

implementations may be combined, supplemented, modified, or removed to produce other implementations. Additionally, one of ordinary skill will understand that other structures and processes may be substituted for those disclosed and the resulting implementations will perform at least substantially the same function(s), in at least substantially the same way(s),

5   to achieve at least substantially the same result(s) as the implementations disclosed. Accordingly, these and other implementations are contemplated by this application and are within the scope of the following claims.

CLAIMS:

1.   A method comprising:

accessing a first depth picture that corresponds to a first video picture (1410);

determining, for a given portion of the first depth picture, a co-located video portion of the first video picture (1420);

accessing a video motion vector indicating motion of the co-located video portion of the first video picture with respect to a second video picture (1420);

accessing a second depth picture that corresponds to the second video picture (1430);

determining a depth portion of the second depth picture, from the given portion of the first depth picture, based on the video motion vector (1430); and

updating the given portion of the first depth picture based on the depth portion of the second depth picture (1435).

2.   The method of claim 1 wherein determining the depth portion of the second depth picture comprises using the video motion vector to determine the depth portion of the second depth picture from the given portion of the first depth picture such that the given portion of the first depth picture is given the same motion as the co-located video portion of the first video picture.

3.   The method of claim 1 wherein the second video picture has a timestamp different from a timestamp of the first video picture.

4.   The method of claim 1, wherein the method is implemented in one or more of a video encoder, a pre-processor for the video encoder, a post-processor for the video encoder, a video decoder, a pre-processor for the video decoder, or a post-processor for the video decoder.

5.   The method of claim 1, wherein updating the given portion of the first depth picture comprises applying one or more of temporal weighted averaging and median filtering to values of one or more of the given portion of the first depth picture or the depth portion of the second depth picture (1425, 1435).

6.   The method of claim 5, wherein respective weights applied to the given portion of the first depth picture and the depth portion of the second depth picture are determined based on one or more of motion criteria and temporal distance criteria (1425).

7.   The method of claim 6, wherein respective weights applied to the given portion of the first depth picture and the depth portion of the second depth picture are determined based on respective amounts of motion in the given portion of the first depth picture and the depth portion of the second depth picture (1425).

8.   The method of claim 7, wherein a value of a particular one of the respective weights increases as the amount of motion decreases (1425).

9.   The method of claim 6, wherein respective weights applied to the given portion of the first depth picture and the depth portion of the second depth picture are determined based on a temporal distance between the first video picture and the second video picture (1425).

10.   The method of claim 9, wherein a value of a particular one of the respective weights decreases as the temporal distance increases (1425).

11.   The method of claim 5, wherein respective weights are applied to the given portion of the first depth picture and the depth portion of the second depth picture on a pixel-basis when the median filtering is applied thereto (1425).

12.   The method of claim 1 further comprising determining the video motion vector.

13.   The method of claim 1 further comprising
       accessing a different video motion vector indicating motion of the co-located video portion of the first video picture with respect to a third video picture;
       accessing a third depth picture that corresponds to the third video picture; and
       determining a depth portion of the third depth picture, from the given portion of the first depth picture, based on the different video motion vector, and

wherein updating the given portion of the first depth picture is further based on the depth portion of the third depth picture as well as the depth portion of the second depth picture.

5   14.   An apparatus comprising:

means for accessing a first depth picture that corresponds to a first video picture;

means for determining, for a given portion of the first depth picture, a co-located video portion of the first video picture;

means for accessing a video motion vector indicating motion of the co-located video

10   portion of the first video picture with respect to a second video picture;

means for accessing a second depth picture that corresponds to the second video picture;

means for determining a depth portion of the second depth picture, from the given portion of the first depth picture, based on the video motion vector; and

15   means for updating the given portion of the first depth picture based on the depth portion of the second depth picture.

15.   A processor readable medium having stored therein instructions for causing a processor to perform at least the following:

20   accessing a first depth picture that corresponds to a first video picture (1410);

determining, for a given portion of the first depth picture, a co-located video portion of the first video picture (1420);

accessing a video motion vector indicating motion of the co-located video portion of the first video picture with respect to a second video picture (1420);

25   accessing a second depth picture that corresponds to the second video picture (1430);

determining a depth portion of the second depth picture, from the given portion of the first depth picture, based on the video motion vector (1430); and

updating the given portion of the first depth picture based on the depth portion of the second depth picture (1435).

30

16. An apparatus, comprising a processor configured to perform at least the following:

accessing a first depth picture that corresponds to a first video picture (1410);

5   determining, for a given portion of the first depth picture, a co-located video portion of the first video picture (1420);

accessing a video motion vector indicating motion of the co-located video portion of the first video picture with respect to a second video picture (1420);

accessing a second depth picture that corresponds to the second video picture (1430);

10   determining a depth portion of the second depth picture, from the given portion of the first depth picture, based on the video motion vector (1430); and

updating the given portion of the first depth picture based on the depth portion of the second depth picture (1435).

15   17. An apparatus comprising:

a temporal filter (330) for accessing a first depth picture that corresponds to a first video picture, and for accessing a second depth picture that corresponds to a second video picture; and

a motion analyzer (310) for determining a video motion vector for a co-located video

20   portion of the first video picture, the co-located video portion of the first video picture corresponding to a given portion of the first depth picture, and the video motion vector indicating motion of the co-located video portion of the first video picture with respect to the second video picture,

wherein the temporal filter determines a depth portion of the second depth picture,

25   from the given portion of the first depth picture, based on the video motion vector, and the temporal filter updates the given portion of the first depth picture based on the depth portion of the second depth picture.

18.   An apparatus comprising:

a temporal filter (330) for accessing a first depth picture that corresponds to a first video picture, and for accessing a second depth picture that corresponds to a second video picture; and

a motion analyzer (310) for determining a video motion vector for a co-located video portion of the first video picture, the co-located video portion of the first video picture corresponding to a given portion of the first depth picture, and the video motion vector indicating motion of the co-located video portion of the first video picture with respect to the second video picture,

wherein the temporal filter determines a depth portion of the second depth picture, from the given portion of the first depth picture, based on the video motion vector, and the temporal filter updates the given portion of the first depth picture based on the depth portion of the second depth picture, and
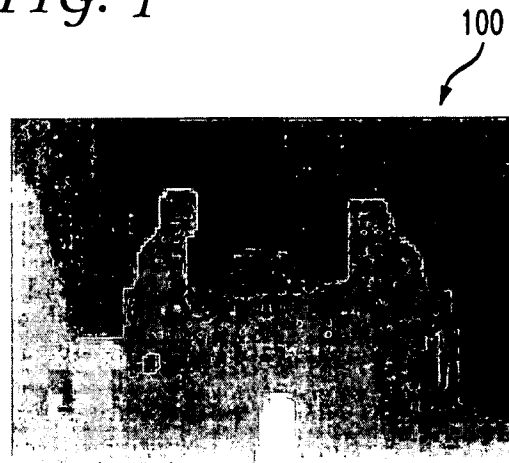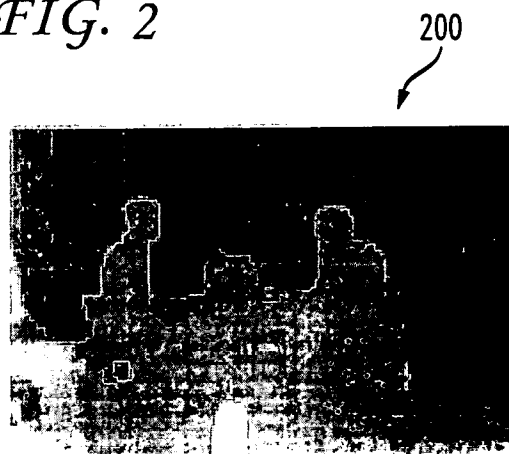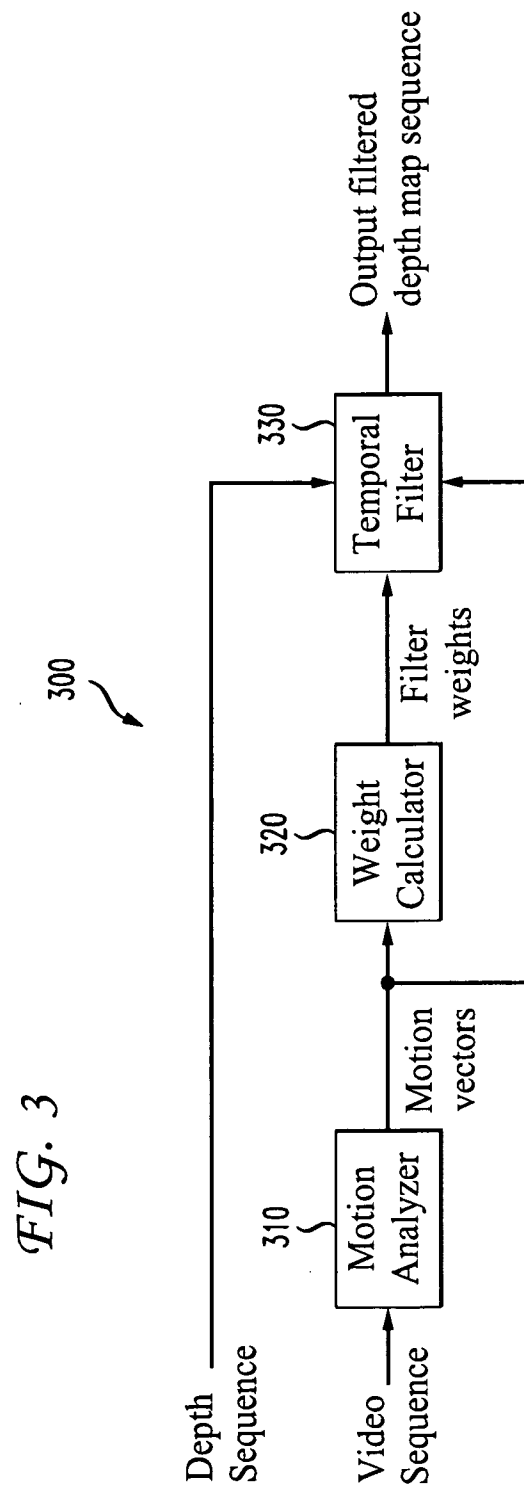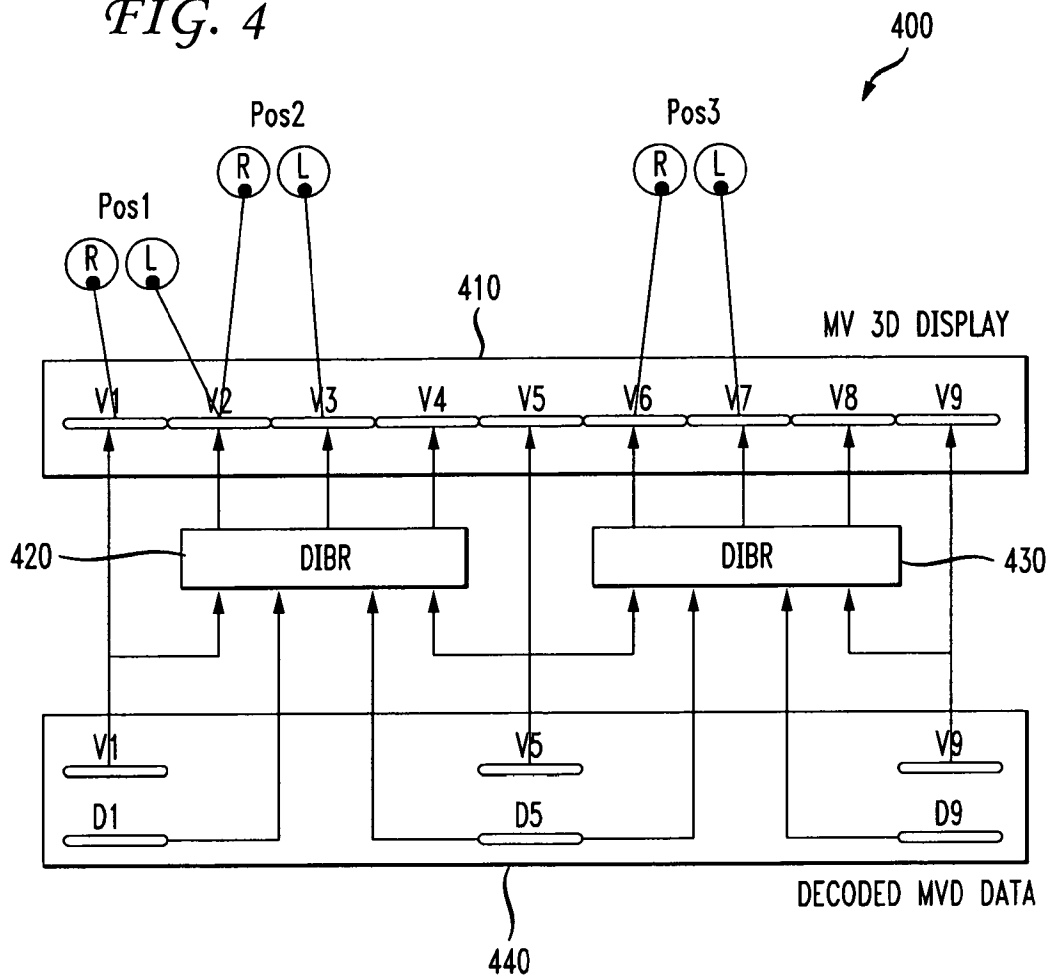
wherein the apparatus further comprises a modulator (520) for modulating a signal that includes the updated given portion of the first depth picture.

19.   An apparatus comprising:

a demodulator (610) for demodulating a signal that includes a first video picture, a second video picture, a first depth picture that corresponds to the first video picture, and a second depth picture that corresponds to the second video picture;

a temporal filter (330) for accessing the first depth picture, and for accessing the second depth picture; and

a motion analyzer (310) for determining a video motion vector for a co-located video portion of the first video picture, the co-located video portion of the first video picture corresponding to a given portion of the first depth picture, and the video motion vector indicating motion of the co-located video portion of the first video picture with respect to the second video picture,

wherein the temporal filter determines a depth portion of the second depth picture, from the given portion of the first depth picture, based on the video motion vector, and the temporal filter updates the given portion of the first depth picture based on the depth portion of the second depth picture.

22

FIG. 1

100

FIG. 2

200

2/9

FIG. 3

3/9

FIG. 4

400

Pos2

Ⓡ Ⓛ

Pos3

Ⓡ Ⓛ

Pos1

Ⓡ Ⓛ

410

MV 3D DISPLAY

| V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 |

420 —

DIBR

DIBR

— 430

V1

V5

V9

D1

D5

D9

DECODED MVD DATA
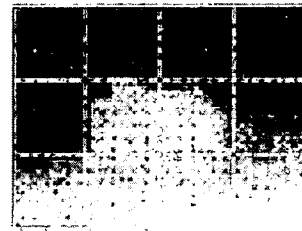
440

FIG. 5

FIG. 6
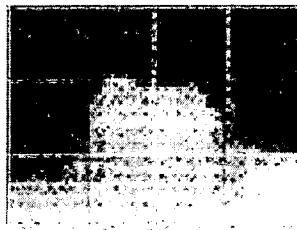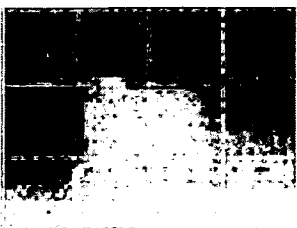
FIG. 7

FIG. 8

Frame 5
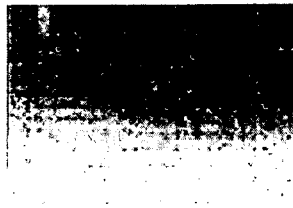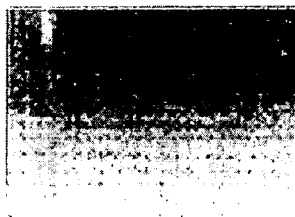
FIG. 9

Frame 10

FIG. 10

Frame 15

FIG. 11

1100
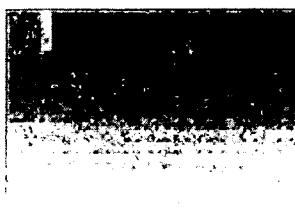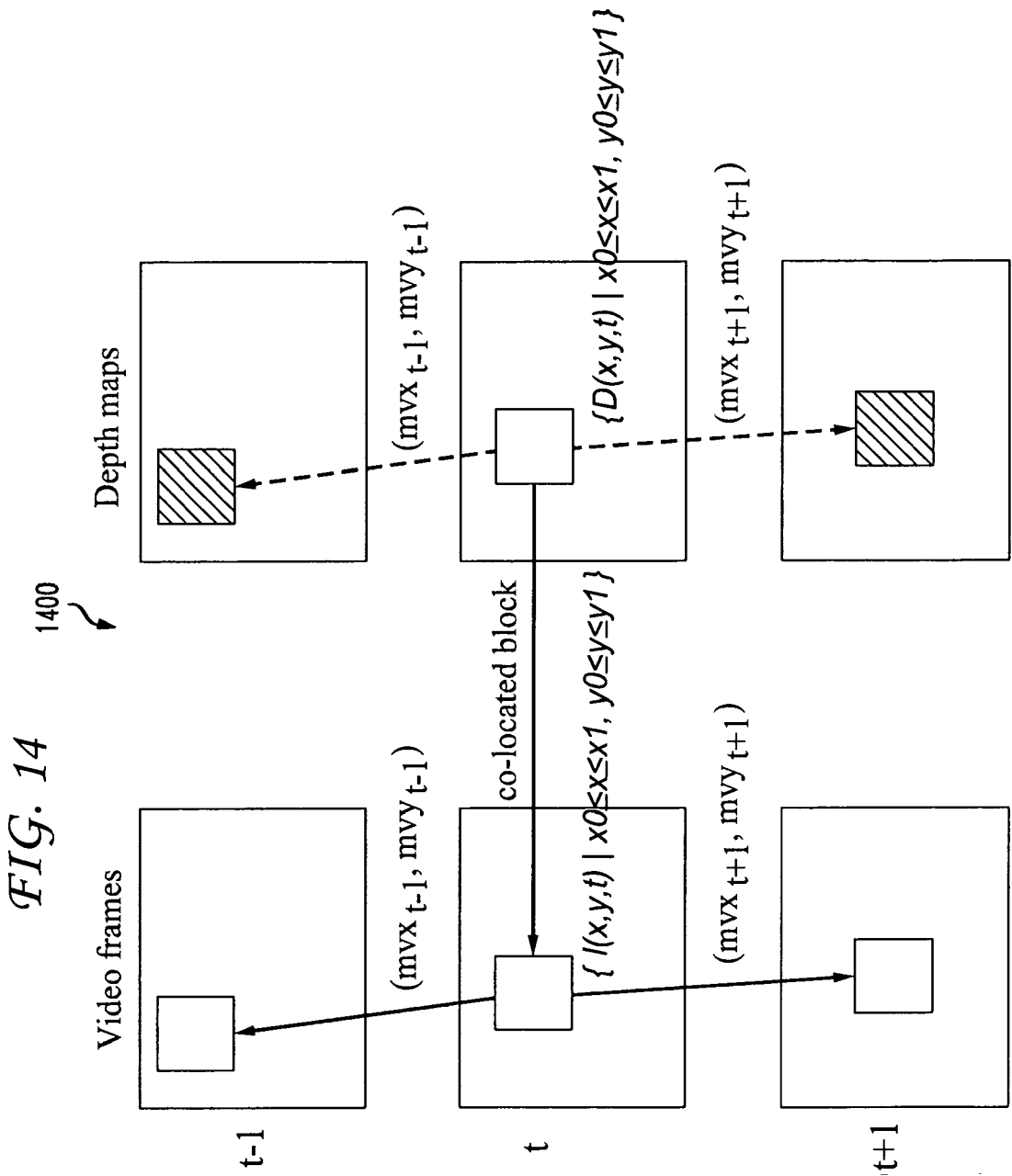
Frame 5

FIG. 12
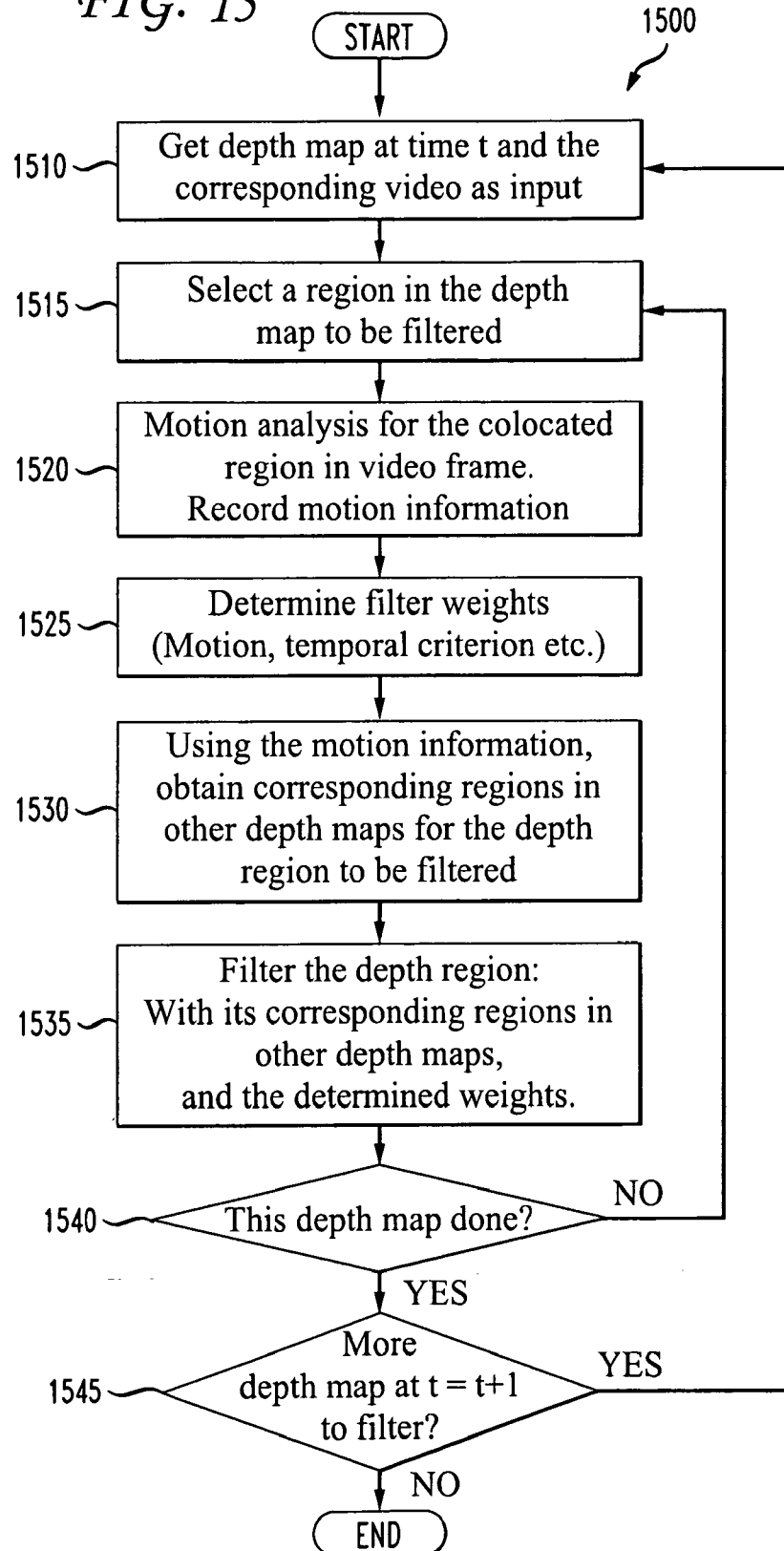

1200

Frame 10

FIG. 13

1300

Frame 15

FIG. 14

1400

Depth maps

Video frames

(mvx_{t-1}, mvy_{t-1})

$\{D(x,y,t) \mid x0 \leq x \leq x1, y0 \leq y \leq y1\}$

(mvx_{t+1}, mvy_{t+1})

co-located block

$\{I(x,y,t) \mid x0 \leq x \leq x1, y0 \leq y \leq y1\}$

(mvx_{t-1}, mvy_{t-1})

(mvx_{t+1}, mvy_{t+1})

t-1

t

t+1

9/9

FIG. 15

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N13/00
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2009/013682 A2 (KONINKL PHILIPS ELECTRONICS NV [NL]; VAREKAMP CHRISTIAAN [NL]) 29 January 2009 (2009-01-29) page 3, line 1 – line 4 page 10, line 24 – line 29 page 11, line 19 – page 14, line 9 _____ -/--	1-19

[X] Further documents are listed in the continuation of Box C.   [X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
7 May 2010	25/05/2010

| Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Fax: (+31–70) 340–3016 | Authorized officer Cooke, Edward |

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	GONG ET AL: "Real-time joint disparity and disparity flow estimation on programmable graphics hardware" COMPUTER VISION AND IMAGE UNDERSTANDING, ACADEMIC PRESS, US LNKD-DOI:10.1016/J.CVIU.2008.07.007, vol. 113, no. 1, 1 January 2009 (2009-01-01), pages 90-100, XP025692173 ISSN: 1077-3142 [retrieved on 2008-08-05] abstract	1,14-19
A	Strecha, Van Gool: "Motion-Stereo Integration for Depth Estimation" Lecture Notes in Computer Science vol. 2351/2002, 1 January 2002 (2002-01-01), pages 495-497, XP002581504 ISSN: 1611-3349 ISBN: 978-3-540-43744-4 Retrieved from the Internet: URL:http://www.springerlink.com/content/gxk86j29936x8ybk/> [retrieved on 2010-05-07] abstract	1,14-19
A	HAI TAO ET AL: "Dynamic depth recovery from multiple synchronized video streams" PROCEEDINGS 2001 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. CVPR 2001. KAUAI, HAWAII, DEC. 8 - 14, 2001; [PROCEEDINGS OF THE IEEE COMPUTER CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION], LOS ALAMITOS, CA, IEEE COMP. SOC, US, vol. 1, 8 December 2001 (2001-12-08), pages 118-124, XP010583736 ISBN: 978-0-7695-1272-3 abstract	1,14-19
A	LEUNG C ET AL: "An energy minimisation approach to stereo-temporal dense reconstruction" PATTERN RECOGNITION, 2004. ICPR 2004. PROCEEDINGS OF THE 17TH INTERNAT IONAL CONFERENCE ON CAMBRIDGE, UK AUG. 23-26, 2004, PISCATAWAY, NJ, USA,IEEE LNKD-DOI:10.1109/ICPR.2004.1333708, vol. 4, 23 August 2004 (2004-08-23), pages 72-75, XP010723866 ISBN: 978-0-7695-2128-2 abstract	1,14-19

-/--

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	YU HUANG ET AL: "A layered method of visibility resolving in depth image-based rendering" 19TH INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, 2008: ICPR 2008; 8 - 11 DEC. 2008, TAMPA, FLORIDA, USA, IEEE, PISCATAWAY, NJ, 8 December 2008 (2008-12-08), pages 1-4, XP031412254 ISBN: 978-1-4244-2174-9 abstract	1,14-19

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2009013682 A2	29-01-2009	EP 2174511 A2	14-04-2010