Tsai and Fan *EURASIP Journal on Image and Video Processing* (2016) 2016:21
DOI 10.1186/s13640-016-0121-x

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Monocular vision-based depth map extraction method for 2D to 3D video conversion

Tsung-Han Tsai* and Chen-Shuo Fan

Abstract

Due to the demand of 3D visualization and lack of 3D video content, a method converting the 2D to 3D video plays an important role. In this paper, a low-cost and high efficiency post processing method is presented to enjoy the vivid 3D video. We present two semi-automatic depth map extraction methods for stereo video conversion. For the static background video sequence, we proposed a method which combined the foreground segmentation with vanishing line technology. According to the separated foreground and background results from foreground segmentation algorithm, the viewer can use their acquired visual experience to initiate this operation with some depth information of background. Moreover, we proposed another conversion method for the dynamic background video sequence. Foreground segmentation was replaced by the relative velocity estimation based on the motion estimation and motion compensation. Combining the depth map from this work and the original 2D video, a vivid 3D video is produced.

Keywords: 3D video, 2D to 3D conversion, Depth map, Foreground segmentation, Motion estimation

1 Introduction

3D video signal processing has become a hot development trend with large potential in visual processing areas. However, the problem of 3D content generation still lingers. Users can just watch computer graphic 3D animations or movie produced by a particular camera setting. Due to the lack of 3D media contents, a technique which converts existing 2D contents into 3D contents can play a key role for the growing 3D markets.

Many methods have been proposed to convert 2D video to 3D video during the past few years. These conversion methods rely on different visual cues ranging from motion information perspective structures. Some methods are based on horizontal parallax. One of the works is based on geometric and texture [1]. The work by Jung et al. [2] proposed a novel line tracing method based on relative height depth cue. The modern methods take advantage of the depth map information to render the stereo and even multiple views for display [3] used depth image-based rendering technique to the display system. Additionally motion estimation technique is aided to cooperate with moving object detection [4, 5] proposed an H.264-based scheme for 2D to 3D video conversion. They used the motion information between successive frames and concerned the depth ambiguity at the boundaries on moving objects. A similar approach on [6] utilized the spatio-temporal analysis of MPEG videos to convert a stereoscopic video.

2D to 3D video conversion method can be divided into two categories according to the situation of human-computer interaction: fully automatic and semi-automatic method [7]. Fully automated method is used to generate 3D video directly from 2D without any human-computer interaction. However, it is always a major issue on fully automatic method to create a robust and stable solution in any general content. This justifies the necessity of human interaction for accurate stereo view generation. By introducing human-computer interactions, semi-automatic methods can balance quality and cost more flexibility than fully automatic methods. Semi-automatic methods can apply with more flexibility than fully-automatic methods. It can keep the quality of stereo view by introducing human-computer interactions. Stereo quality and conversion cost are determined by the key frame intervals and the accuracy

* Correspondence: han@dsp.ee.ncu.edu.tw
Department of Electrical Engineering, National Central University, No. 300, Jung-Da Rd., Jung-Li City, Taiwan 320, Republic of China

of depth maps on key frames. More accurate depth map will improve the stereo quality, but increase the conversion cost as well. Therefore, a tradeoff has to be made in order to obtain satisfactory quality at an acceptable cost.

In this paper, the video conversion method for 2D to 3D video is proposed. We apply the semi-automatic depth map extraction approach to provide the high-quality stereo video as a 3D entertainment. Two methods are constructed to deal with static and dynamic background scenes, respectively. The organization of this paper is described as follows. We provide the related works on Section 2. In Section 3, we present an overview of the proposed method. In Section 4, Method-1 is introduced with Gaussian mixture model for background and moving object detection. In Section 5, Method-2 is introduced with a relative velocity estimation method for moving object detection. In Section 6, we present the depth extraction and depth fusion process where both methods are adapted. Visual results and comparison data are shown in Section 7, and a conclusion is given in Section 8.

2 Related works

This section introduces the related works on fully automatic and semi-automatic, which are the two main methods for 2D to 3D video conversion. Referring to fully automated method, Knorr et al. in [8] proposed a geometric segmentation approach for dynamic scenes. It included a prioritized sequential algorithm for sparse 3D reconstruction and camera path estimation to efficiently reconstruct 3D scenes from broadcasting video. In [9], a structure was proposed from motion method to automatically recover 3D structure of the scene. However, this method has some limitations in camera movement and scene movement and thus reduces the wide availability of this method. In [10], Zhang et al. recovered consistent video depth maps and proposed a novel method based on bundle optimization. Zhang et al. [11] used a method which mainly integrates occlusion and visual attention organically to calculate depth map. Recently, Lei et al. [12] proposed example-based video stereolization with foreground segmentation and depth propagation according to the key and non-key frames in 2D videos.

With respect to semi-automatic method, it is widely proposed in recent years. Guttmann et al. [13] presented a semi-automatic system which propagates a sparse set of disparity values across a video. It employed classifiers combined with solving a linear system of equations and only requires sparse set of disparity values on the first and last frames of the video clip for reducing manual labor. Yan et al. [14] presented an effective method to semi-automatically generate high-quality depth maps for monocular images based on limited user inputs and depth propagation. They specify the depth values of the selected pixels and locate the approximate positions of T-junctions by user inputs and then generate depth maps by depth propagation combining user inputs, color, and edge information. For the purpose of stereoscopic 3D conversion, Phan et al. [15] proposed the module to alleviate much user input, as only the first frame needs to be marked. As in semi-automatic conversion approach, we had some previous result for static background scene [16]. The main concept for this design is based on the vanishing point detection for depth map realization. As discussed in [17], a scene with vanishing line should be the most representative and easy to manipulate. This simple technique often leads to representative result especially for some man-made environments where they are always presented with many regular structures and parallel line.

3 Overall of proposed method

3.1 System overview

In this paper, we propose the techniques where scene geometry cues from a video and incorporate them into video segmentation and grouping. By inferring the depth information, some monocular depth cues have been proposed such as texture variations gradients, haze, and defocus [18]. As a semi-automatic concept, we manipulate it by setting five initial points. Among them, four points can induce two vanishing lines. Following these two vanishing lines, the vanishing point is cooperated with the fifth point to decide the horizontal line and derive the depth map.

Additionally, our proposed technique includes two methods to cover all the video scene situations. Method 1 is constructed for static background video sequence. Referring to Method-2, a more elaborate mechanism is proposed for the dynamic background video sequence. The whole system is illustrated in Fig. 1. First, it initiates a scene detection to classify the characteristic of background scene. In this scene detection, we apply Gaussian distribution model to analyze the first frame and set it as background frame. Then, the consecutive frames are operated with the difference of the background frame. The detection equation is as follows:

$$S(x,y) = \begin{cases} 1, & |I(x,y)-R(x,y)| \le T \\ 0, & |I(x,y)-R(x,y)| > T \end{cases} \quad (1)$$

where $I(x, y)$ denotes the current frame and $R(x, y)$ denotes the reference frame, respectively. T is a threshold set as one. If the number of one in $S(x, y)$ is more than 2/3 of the total amount, the characteristic is detected as dynamic since more pixels are varied; otherwise, the scene is detected as static scene.

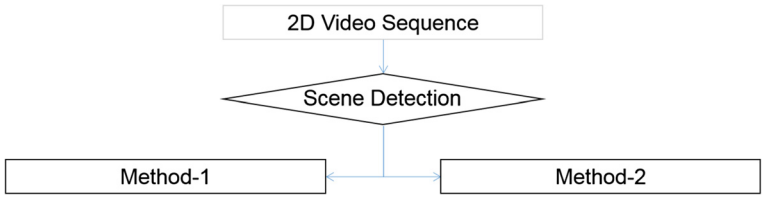

Fig. 1 The whole system block

3.2 Overview on Method-1

For Method-1, we propose a novel semi-automated approach based on the input from the user to specify a set of initial conditions. First, the vanishing line extraction [19] is used in these static background video sequences. To generate depth maps of these scenes, three key issues are addressed. First is the depth layers acquisition of the static scenes. Second is the precise segmentation of moving objects. Third is the depth assignment to the segmented objects. By precise segmentation on the foreground and background, the separated scenes can be fused with the corresponding depth map.

The abstract structure of the proposed Method-1 is depicted in Fig. 2. Dotted line means the information is only used for initial condition, and dotted block means the function is only used for the first used frame.

3.3 Overview on Method-2

Dynamic background contains time-varying information and clutter-like appearance of dynamic textures [20]. Because this kind of video scene needs more consideration than the static background video scene, we propose Method-2 on the dynamic background for video sequence conversion. Traditionally, background modeling and subtraction methods [21] have a strong assumption where the scenes of static structures are with limited perturbation. These methods will perform poorly in dynamic scenes. Here, we apply the relative velocity method to segment foreground from the dynamic background. The abstract structure of the proposed Method-2 is depicted in Fig. 3.

3.4 The advantages of the proposed method

Three main concepts are used in our proposed method. The first is that moving objects were always being the focus of video viewer. The second is that people can use their acquired monocular depth to directly provide key hints on computer depth generation. Then, the computation time of background classification and monocular depth cue estimation could be saved. For saving operating time, the third concept is that the user will not frequently interact with this conversion system. According to the above concepts, the segmented result of background and foreground and a user-guided vanishing line extraction method are combined with motion information between neighbor frames.

4 Static background segmentation

Background modeling and moving object detection are based on the adaptive background subtraction method. In this method, each pixel is modeled as a mixture of Gaussians with the on-line approximation to update the model [22]. The Gaussian distributions are then evaluated to determine which are most likely to the result from a background process.

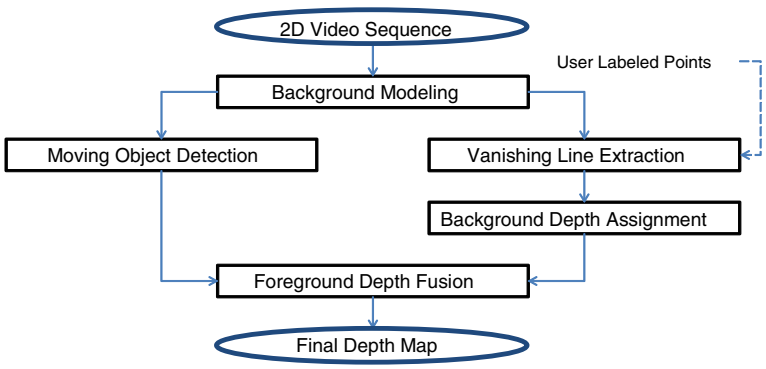

Fig. 2 Framework of the proposed Method-1

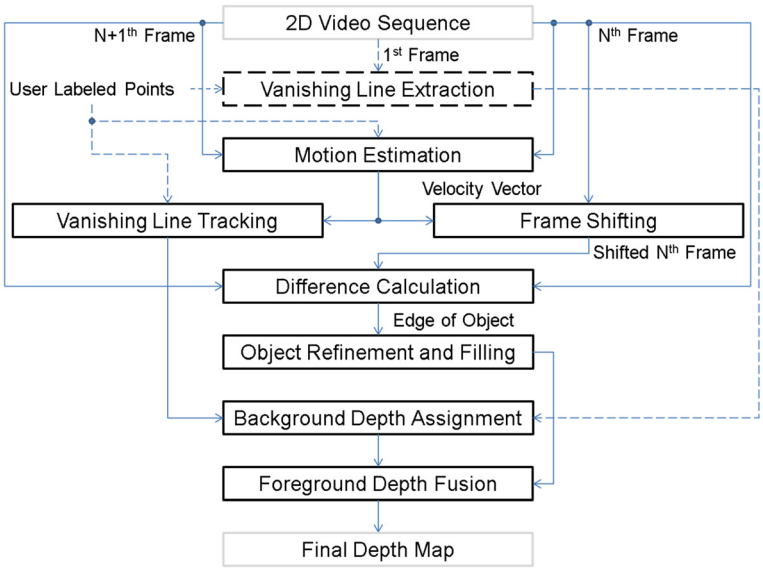

Fig. 3 Framework of the proposed Method-2

4.1 Background modeling

For a certain pixel in a frame, at any time t, the set of pixel values can be denoted as $X = \{X_1,\ldots,Xt\}$. The recent history of each pixel is modeled by a mixture of K Gaussian distributions. The probability of observing the current pixel value is as

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta\left(X_t, \mu_{i,t}, \sum_{i,t}\right) \tag{2}$$

K is the number of distributions. $\omega_{i,t}$ and $\mu_{i,t}$ are an estimate of the weight and the mean value of the ith Gaussian in the mixture at time t, respectively. $\Sigma_{i,t}$ is the covariance matrix of the ith Gaussian in the mixture at time t. η is the Gaussian probability density function as the following:

$$\eta\left(X_t, \mu, \sum\right) = \frac{1}{(2\pi)^{\frac{n}{2}}\left|\sum\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu_t)^T \sum^{-1}(X_t-\mu_t)} \tag{3}$$

Based on the persistence and the variance for each of the Gaussians of the mixture, we determine the Gaussians which may correspond to the background colors. Because there is a mixture model for every pixel in the image, we refer the version of [23] and modify it to improve the convergence speed and model accuracy. Every new pixel value X_t is checked against the existing K Gaussian distributions until a match is found. A match is defined as a pixel value within 2.5 standard deviation of a distribution. If one of the K distributions matches the current pixel value, the parameters, e.g., weight, learning factor, covariance, mean value of the distribution, are updated. Figure 4a, b shows an example with the original and the background modeling result.

4.2 Moving object detection

After the background modeling of video frame, Gaussians are ordered by the value of ω/σ, representing the ratio of weight over covariance of distribution. For the ith background model where the largest value of ω/σ is, moving objects are separated from the original 2D video. Then binarize is made to extract the background and the moving objects, i.e., pixel value of the background is

Fig. 4 Moving object detection results. **a** The original video. **b** The result for background modeling. **c** Moving object

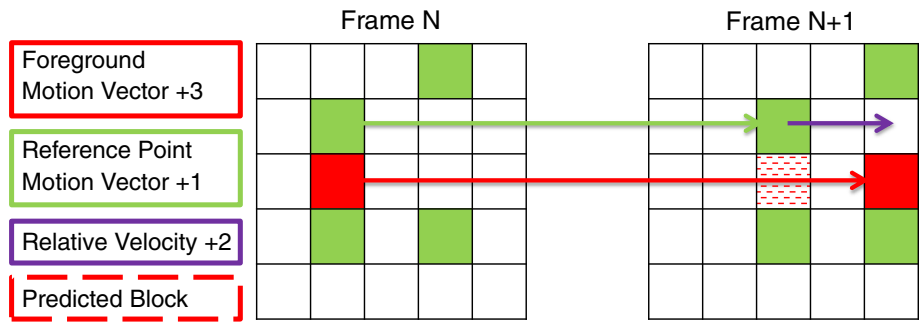

Fig. 5 Representation of relative velocity

set to 0 and pixel value of the moving objects is set to 255. The separation process is described as follows:

$$A(x,y) = \begin{cases} 255, & I(x,y) - \mu_k \leq T \\ 0, & I(x,y) - \mu_k > T \end{cases} \quad (4)$$

where $A(x,y)$ denotes the binarization result of moving object detection. $I(x,y)$ denotes the current pixel value of image. T is a threshold that compares with the difference between the current pixel and ith Gaussians mean values. T is also estimated depending on the specified probability PR [24]. Figure 4c shows the result of moving object detection.

5 Dynamic background subtraction

In this paper, foreground is defined as an object which has different motion vector with respect to the most similar motion vectors between each neighbor frames. Relative velocity is computed by this motion vectors and adapted to classify foreground and background, as shown in Fig. 5. Also, we only use the luminance value of YUV color space for frame shifting and vanishing line tracking.

5.1 Motion estimation on relative velocity

Motion vectors are extracted from block motion estimation where a full search algorithm [25] is implemented. Search range and block size are defined for a tradeoff between saving computation time and precise prediction. By two adjacent frames, the block of current frame is compared with the block which has minimum sum of absolute difference (SAD) of the reference frame [26]. Currently, high efficiency video coding (HEVC) is developed to provide better efficiency than previous video coding standards. The high coding efficiency is suitable for deal with sophisticated motion estimation, but with the high efficiency, the price is higher computational complexity [27, 28].

Relative velocity estimation is developed based on the motion estimation. When the first frame of the video sequence is input, the user will assign at least four points which are on the paths of vanishing lines for initial condition setting. After the vanishing lines of the first frame are extracted, the second frame of the video would track these user-guided points by motion estimation and reconstruct this vanishing lines. Thus, the points of each vanishing line in the current Nth frame is tracked from the reference $N+$

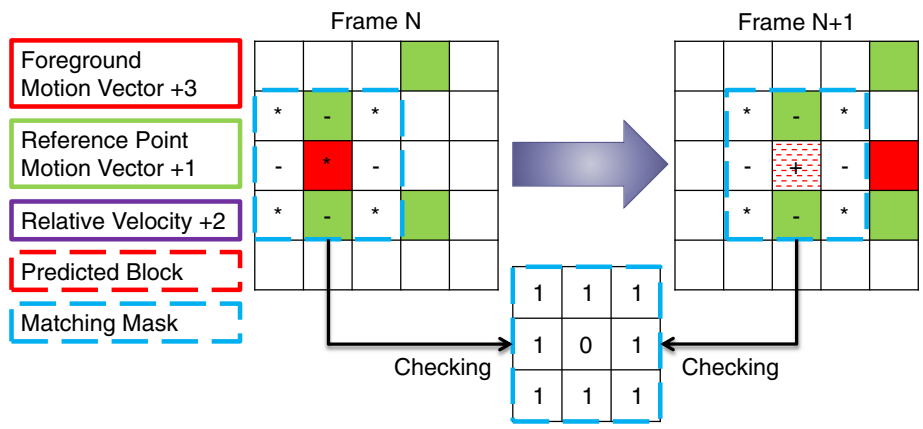

Fig. 6 Representation of foreground contour refinement

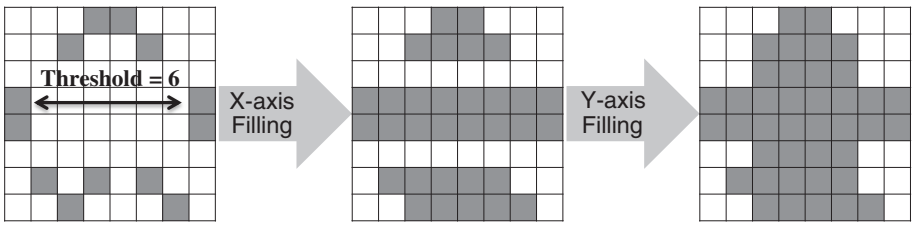

Fig. 7 Processing steps of object filling.

1th frame. If tracked points are occluded by the foreground, the mistake of the tracked luminance value will be emerged on the foreground. It will induce a wrong tracking path from those feature points. Therefore, we set an occlusion prevention function in (5) to restrain the shifting value of points when the feature points are suddenly shifting at large number or a small number.

$$\begin{cases} \text{TS} + 5 > \text{MV of feature points} > \text{TS-5}, \text{ shift value=MV} \\ \text{Otherwise, shift value} = 0 \end{cases} \tag{5}$$

In each different situation of background velocity, those shift number of assigned feature points will be recorded and compared to calculate the statics of average shift number. Thus, the function (5) is dynamically changed with the appropriate shift number within the threshold and the range. Because the user-labeled points are usually seemed as a part of background, the mean value of motion vectors of these feature points can be treated as a master background pixel shifting value. Then, the original Nth frame will be shifted and assumed as the predicted $N+1$th frame.

5.2 Moving object extraction

After the original Nth frame is shifting, the predicted $N+1$th frame is subtracted with original $N+1$th frame of the video sequence. Then, an object is extracted since it has relative slow or fast motion velocity to the most background pixels. If the difference of the pixel is over a threshold value, this pixel will be assigned as 255 of luminance value; otherwise, the pixel will be assigned as 0 of luminance value. Once the camera is moved, the most background pixels will induce a steady motion vector. By calibrating this steady motion vector, the dynamic background can be assumed as static background. At the same time, unstable moving objects are emerged because of variation motion vectors. We use (6) to distinguish the difference of luminance and detect the foreground.

$$\begin{cases} P(x,y) - O(x,y) > TF, F(x,y) = 255 \\ \text{Otherwise, } F(x,y) = 0 \end{cases} \tag{6}$$

$P(x,y)$ means the pixel value of the predicted frame from the frame shifting block and $O(x,y)$ means the pixel value of the original frame. $F(x,y)$ means the frame to be converse at once. According to Fig. 6, we use (7) to refine the extracted foreground, where mask accumulation is the sum total from background mask.

$$\begin{cases} \text{Mask Accumulation} > TM, F(x,y) = 255 \\ \text{Otherwise}, F(x,y) = 0 \end{cases} \tag{7}$$

5.3 Object filling

As mentioned above, 255 of luminance value is produced by two kinds of situation. One is that the

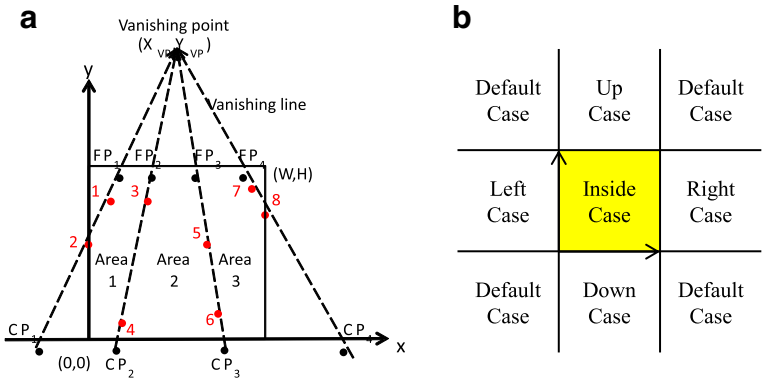

Fig. 8 a User-defined points of video frame. **b** Location of vanishing point heuristic rules

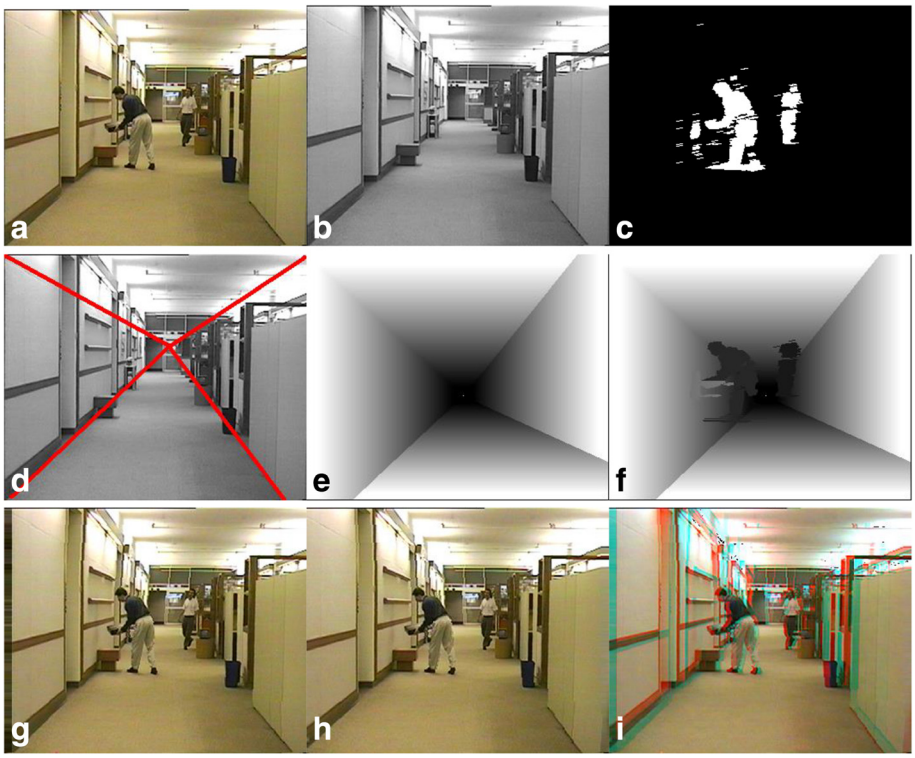

Fig. 9 Simulation results of Method-1 for the 100th frame. **a** The original video frame. **b** Background modeling result. **c** Foreground detection result. **d** User-defined vanishing line extraction on background modeling. **e** Gradient depth plane of background. **f** Depth fusion between background depth map and foreground results. **g** Left view result. **h** Right view result. **i** Anaglyph synthesized of left and right view

pixel is not at the predicted location of the $N+1$th frame. The other is that the pixel is at the unpredictable location. Two white pixels are emerged at the same time and brought to be a set. However, the mean value of motion vectors is useful to distinguish the correct foreground pixel location from inverse checking, as shown in Fig. 6. According to the location of each white pixel set, if the summation of the surrounding filtered pixels of the $N+1$th frame matches the summation of surrounding filtered pixels of Nth frame, the white pixel can be kept; otherwise, the white pixel will be assigned as 0 of luminance value. White pixels represent the edge information of moving objects. Foreground is extracted by filling 255 of luminance value from the left edge to the right edge. Figure 7 shows the procedures of object filling.

6 Depth extraction and depth fusion process

We generate the depth map information to exploit the line features from a single frame. This is used as an input source of extracting geometric cue, i.e., vanishing point (VP). Most scenes are composed of parallel lines. Parallel lines in real scenes appear to converge with distance in perspective image, eventually reaching a VP at the horizon. Linear perspective is an important depth cue for these scenes. Taking into account the information collected in the pre-process analysis, a series of intermediate steps are used to recover the final depth map. These steps can be resumed in vanishing line extraction, gradient planes generation, and depth gradient assignment.

6.1 Vanishing line extraction and gradient plane generation

We adopt user-defined method to extract the main vanishing line. Each vanishing line is consisted of two points. Users are required to identify these two points which are on the estimated vanishing line path. The estimation on main vanishing lines is based on common sense, i.e., edge of the road or wall. These two points are marked as the far to near one according to the reconstruction background scene. As shown in Fig. 8a, red point shows the labeling order. Then, VP and some vanishing lines are computed by these red points. Black point shows the intersection of estimated vanishing line and boundary of the video frame.

During this step, the position of the computed VP in the video frame is analyzed. Considering to the

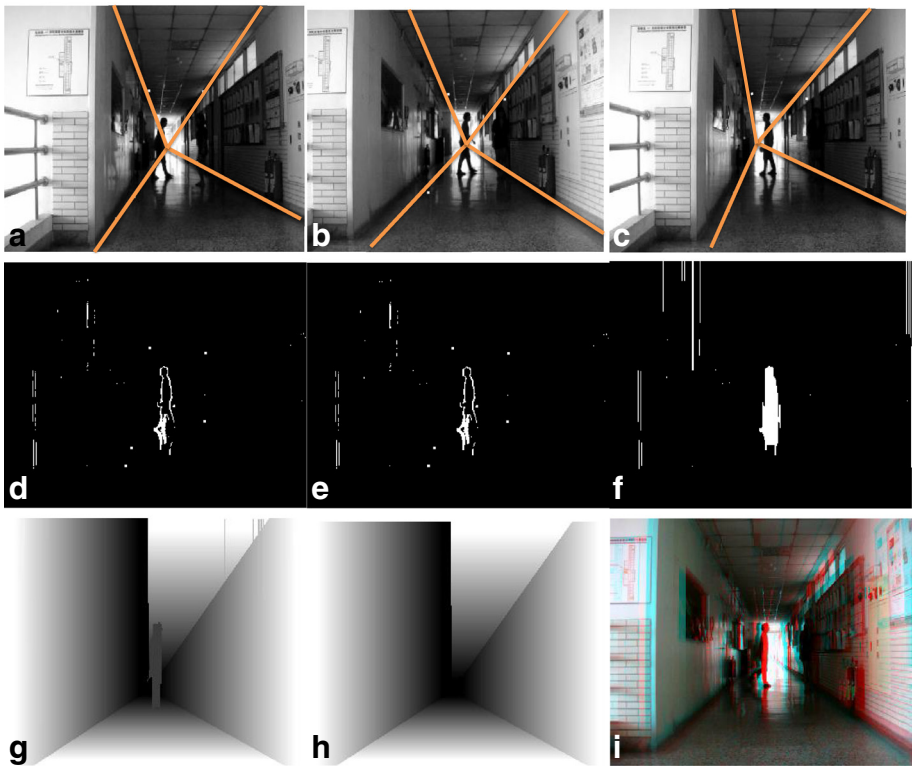

Fig. 10 Simulation results of Method-2 for the 408th frame. **a** Tracking and vanishing line extraction of the 408th frame. **b** Tracking and vanishing line extraction of the 590th frame. **c** Tracking and vanishing line extraction of the 691th frame. **d** Difference calculation between the 690th frame and the 691th frame. **e** Foreground contour refinement of the 691th frame. **f** Foreground filling of the 691th frame. **g** Background depth assignment. **h** Foreground depth fusion. **i** Anaglyph synthesized view

location of vanishing point, six different cases are distinguished as

1) Up case: $0 \le X_{vp} \le W \ \cap \ Y_{vp} \ge H$
2) Down case: $0 \le X_{vp} \le W \ \cap \ Y_{vp} \le 0$
3) Right case: $X_{vp} \ge W \ \cap \ 0 \le Y_{vp} \le H$
4) Left case: $X_{vp} \le 0 \ \cap \ 0 \le Y_{vp} \le H$
5) Inside case: $0 < X_{vp} < W \ \cap \ 0 < Y_{vp} < H$
6) Default case: Otherwise above 5 cases

where H and W are the height and width of image size, respectively. For each case, a set of heuristic rules based on the first computed VP allows to generate the gradient plane which is set by depth variation. Figure 8b shows six cases where the yellow area means the video frame location.

6.2 Background depth extraction

Background depth level is assigned to every pixel depth of gradient planes. The default depth range is from 0 to 255. Two main assumptions are used.

1. Higher depth level corresponds to lower grey values.
2. The VP is the most distant point from the observer.

For every two neighboring estimated vanishing lines, a series of depth planes are assigned. For the up-case in

Fig. 11 Experimental results for static background video

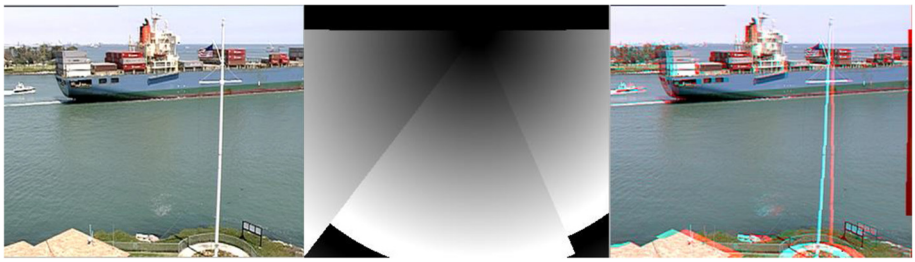

Fig. 12 Experimental results for dynamic background video

Fig. 8a, there are two kinds of intersections, CP_n and FP_n. CP_n means the closer point from video viewer's vision on the nth vanishing line, and FP_n means the farther point. For a pixel $I_n(i,j)$ in nth area, its depth value is given by (8), (9), (10), and (11).

$$L_p = \max\left\{ \mathrm{FP}_n\bar{\mathrm{F}}\mathrm{P}_{n+1} \quad | n = 1, 2, \ldots N \right\} \tag{8}$$

$$R = W^{'} \times \frac{LL}{255} \tag{9}$$

$$W_n = \frac{\mathrm{FP}_n\bar{\mathrm{F}}\mathrm{P}_{n+1}}{L_p} \tag{10}$$

$$\mathrm{depth} = W_n \times \frac{|I_n\bar{\mathrm{V}}\mathrm{P} - \mathrm{FP}_n\bar{\mathrm{V}}\mathrm{P}|}{R} \tag{11}$$

L_p is the most distant one from each two continuous farther points of p area. LL is a length computed by two middle points; one middle point is within FP_p and FP_{p+1}, and the other is within CP_p and CP_{p+1}. R is a depth range proportioned to the distance of video frame. $W^{'}$ stands for user-defined weighting factor when LL is smaller than 255. W_n stands for the weighting factor to assign depth level relative to the nth area. The depth layers of the static background scene are extracted through this way.

In each case, distance between VP and the most distant intersections of boundary of video frame and vanishing line will be considered as a proportion to the depth range. Regarding to the depth assignment of inside-case, it is combined with four cases except the default case. In terms of default case, the boundary of video frame with the most CP will be only considered as four selecting cases except the inside case. Human vision is more sensible to deep variations for close objects than for far ones. In other words, faster vanishing-line convergence induces higher deep variation. Thus, the deep levels have an increasing slope from the closest position to the farthest VP.

6.3 Moving object depth extraction

We observe that the plane of moving object is belonged to one single depth value. It is obvious that an object depth on the plane is determined by the gravity direction which is perpendicular to the ground. Thus, we extract the pixels which form moving object's base and calculate the average depth value of these pixels in the background depth map as the following:

$$D = \mathrm{avg}(d_1 + d_2 + \ldots + d_n) \tag{12}$$

where D and d_n are the depth values of the moving object and the depth value of each pixel that forms the moving object's base.

7 Experimental results

7.1 Simulation results

Figure 9 shows the result of Method-1 with its detailed processing steps. We use "Hall" of 352 × 288 frame size as the video test sequence. In Fig. 9d, the user begins to label points which are on the paths of estimated vanishing lines.

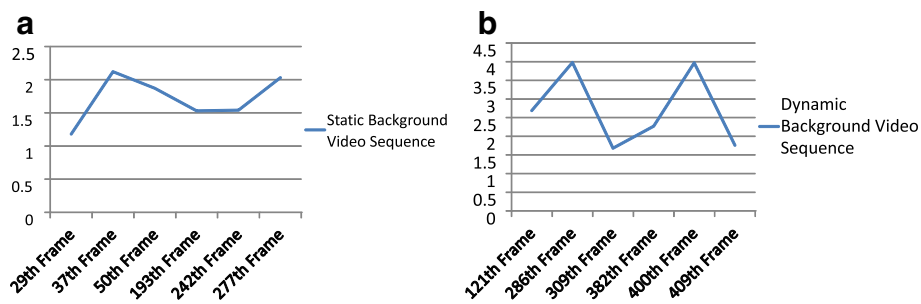

Fig. 13 MSE comparison of two kinds of video sequence. **a** Static background video. **b** Dynamic background video

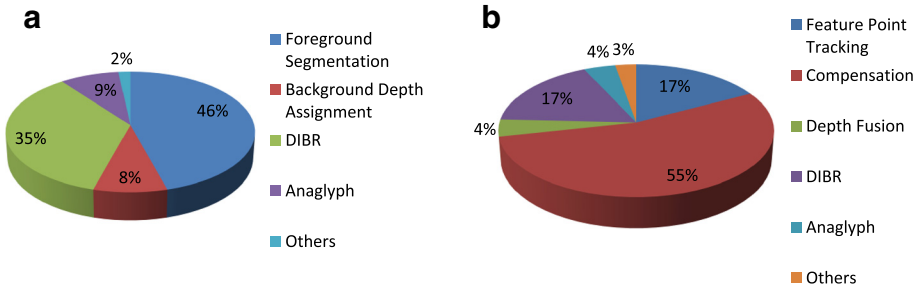

Fig. 14 Execution time profiling. **a** Method-1. **b** Method-2

The red lines are connected by six user-defined points. The computed VP is located at the middle of the video frame. Consequently, the depth gradient plane generation is classified as inside-case and the generated results are shown from Fig. 9g to i.

Referring to Method-2, Fig. 10 shows the result with its detailed processing steps. We use 352×288 frame size as the test video sequence with dynamic foreground moving objects in the dynamic background scene. At first, the users begin to label points which are on estimated vanishing lines path from the first frame of the test sequence. Figure 10a–c shows a series of tracking results. Orange lines are on behalf of extracted vanishing lines. Figure 10d–f shows the process of foreground

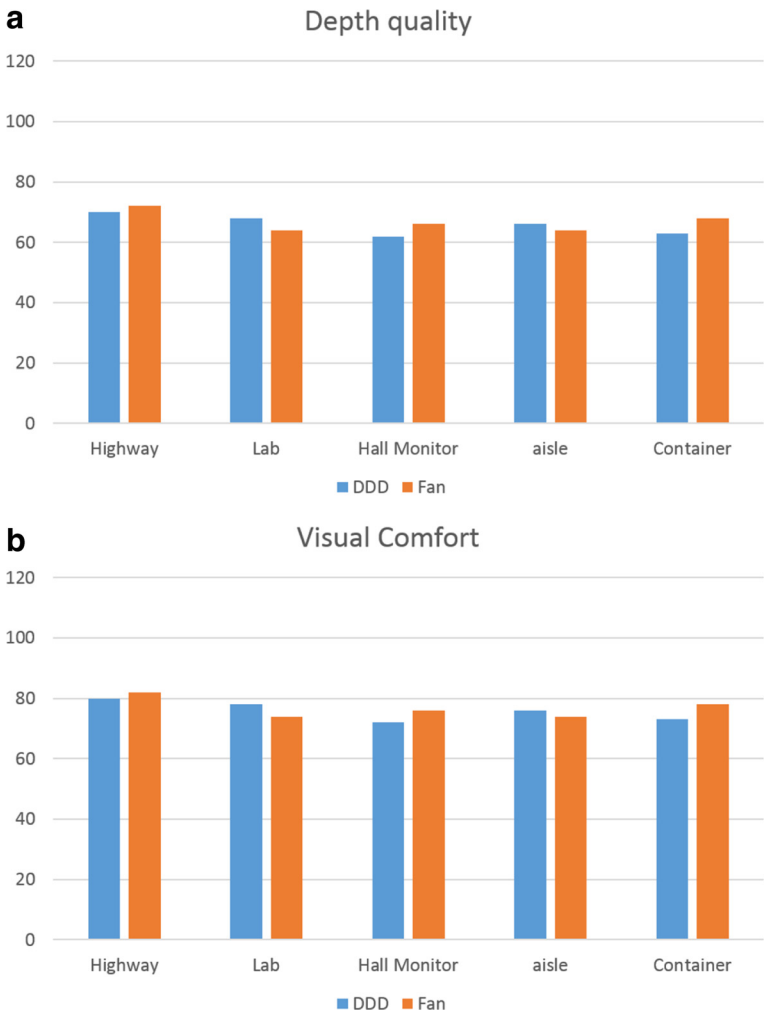

Fig. 15 Subjective evaluation results for **a** depth quality and **b** visual comfort

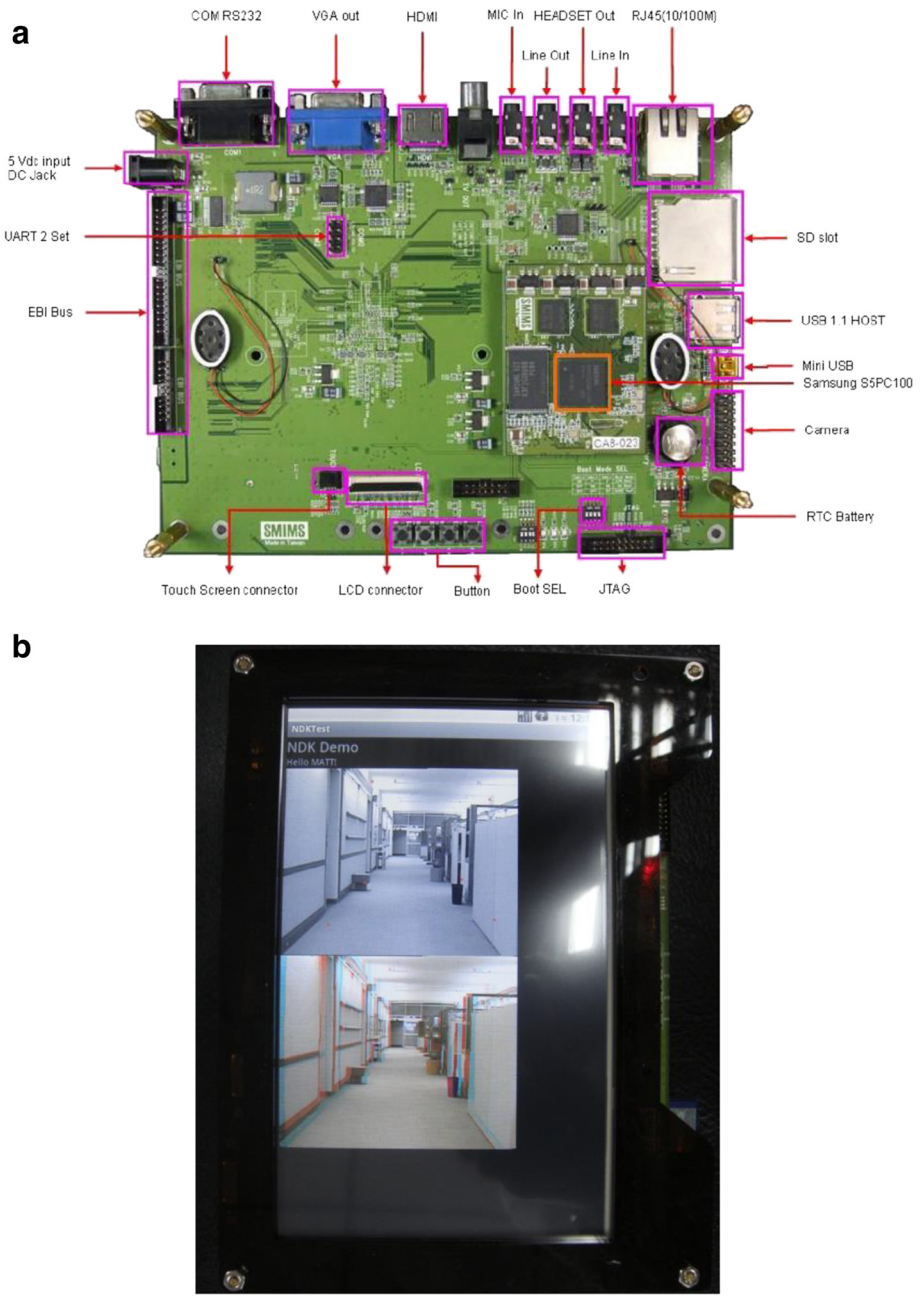

Fig. 16 **a** SMIMS VeriSoC-CA8 platform. **b** The demo view result

segmentation. In Fig. 10g–i, they indicate the depth generation and final stereo video generation results. The gradient plane generation of this test sequence is classified as inside-case.

The other simulation results are shown in Figs. 11 and 12 with static and dynamic backgrounds, respectively. Briefly, we only show the depth map result and the synthesized view. Furthermore, we present the MSE comparison results of video in Fig. 13 with two situations. The dynamic background video sequence has higher MSE value than the static video sequence because the relative velocity is much more unstable and more multiple directional features are included.

7.2 Evaluation and implementation results

We implement our method on a workstation equipped with Intel 2.4GHz Core2 Quad CPU Q6600 and 2GB RAM. To evaluate the performance fairly, the entire algorithm is run with single thread. Every test sequence is restricted to 352 × 288 frame size. The profiling result is

shown in Fig. 14. For Method-1, the foreground segmentation consumes nearly half computation time. This is because several complex functions are included such as area filter and labeling. For Method-2, the compensation occupies more than half of the computation time. Figure 15 shows the evaluated visual comfort of the proposed algorithm with a reference design DDD commercial tool [29].

We also implement our design as a real demo system on SMIMS VeriSoC-CA8 with Android OS. We use the Java Native Interface (JNI) technique to implement Java call C or C call Java for data transferring. Most of the modules are described by C++ and the other communications, e.g., touch panel, SD card reader, LCD display, are described by Java. JNI technique can co-work with C ++ and Java, and thus, it can execute faster than pure Java program. Figure 16 shows the peripheral of SMIMS-CA8 development platform and the demonstration result.

8 Conclusions

In this paper, a monocular vision-based depth map extraction method for 2D to 3D video conversion is presented. We proposed a low complexity and high integration mechanism and also concern the characteristic in sequence. First of all, we initiate a scene detection to classify the characteristic of the background scene. Then, we provide two conversion methods for static and dynamic backgrounds, respectively. By the semi-automatic vanishing line extraction method, it can save much computation time and increase the precision of vanishing line detection. The estimated depth map can be used to generate the right and left view images for each frame to generate a 3D video result. It indicates that the proposed framework can process 2D to 3D conversion with high quality result.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

T-HT carried out the algorithm studies, participated in the simulation, and drafted the manuscript. C-SF carried out the platform implementation and helped to draft the manuscript. All authors read and approved the final manuscript.

Acknowledgements

This research was supported by the Ministry of Science and Technology, Taiwan, under Grant 104-2220-E-008-001.

Received: 14 October 2015 Accepted: 26 April 2016
Published online: 03 June 2016

References

1. K Han, K Hong, Geometric and texture cue based depth-map estimation for 2D to 3D imageconversion, IEEE International Conference on Consumer Electronics (ICCE), 651–652 (2011). doi:10.1109/ICCE.2011.5722790
2. YJ Jung, A Baik, J Kim, D Park, A novel 2D-to-3D conversion technique based on relative height depth cue. Proc. SPIE **7234**, 72371U-1–72371U-8 (2009)
3. R Liu, W Tan, YJ Wu, YC Tan, B Li, H Xie, G Tai, X Xu, Deinterlacing of depth-image-based three-dimensional video for a depth-image-based rendering system. J. Electron. Imaging. **22**(3), 033031 (2013)
4. XJ Huang, LH Wang, JJ Huang, DX Li, M Zhang, A depth extraction method based on motion and geometry for 2D to 3D conversion, in *Third International Symposium on Intelligent Information Technology Application (IITA)*, vol. 3, 2009, pp. 294–298. 21–22 Nov. 2009
5. MT Pourazad, P Nasiopoulos, RK Ward, An H.264-based scheme for 2D to 3D video conversion. IEEE Trans. Consum. Electron. **55**(2), 742–748 (2009)
6. GS Lin, HY Huang, WC Chen, CY Yeh, KC Liu; WN Lie, A stereoscopic video conversion scheme based on spatio-temporal analysis of MPEG videos. Eur. J. Adv. Sign. Process. DOI:10.1186/1687-6180-2012-237
7. Z Li, X Cao, X Dai, *A novel method for 2D-to-3D video conversion using bi-directional motion estimation*. Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference, 2012
8. S Knorr, E Imre, B Ozkalayci, A Alatan, T Sikora, A modular scheme for 2D/3D conversion of TVbroadcast Proc. 3DPTV. 703–710 (2006). doi:10.1109/3DPVT.2006.15
9. T Huang, A Netravali, Motion and structure from feature correspondences: a review. Proc. IEEE **82**(2), 252–268 (1994)
10. G Zhang, J Jia, T Wong, H Bao, Recovering consistent video depth maps via bundle optimization. Proc. CVPR. 1–8 (2008). doi:10.1109/CVPR.2008.4587496
11. J Zhang, Y Yang, Q Dai, A novel 2D-to-3D scheme by visual attention and occlusion analysis. 3DTV Conf. 1–4 (2011). doi:10.1109/3DTV.2011.5877189
12. L Wang, C Jung, Example-based video stereolization with foreground segmentation and depth propagation. Multimed. IEEE Trans. **16**(7), 1905–1914 (2014)
13. M Guttmann, L Wolf, D Cohen-or, Semi-automatic stereo extraction from video footage. Proc. ICCV. 136–142 (2009). doi:10.1109/ICCV.2009.5459158
14. X Yan, Y Yang, G Er, Q Dai, Depth map generation for 2D-to-3D conversion by limited user inputs and depth propagation. 3DTV Conf. 1–4 (2011). doi:10.1109/3DTV.2011.5877167
15. R Phan, D Androutsos, Robust semi-automatic depth map generation in unconstrained images and video sequences for 2D to stereoscopic 3D conversion. Multimed. IEEE Trans. **16**(1), 122–136 (2014)
16. TH Tsai, CS Fan, CC Huang, *Semi-automatic depth map extraction method for stereo video conversion*. 2012 Sixth International Conference on Genetic and Evolutionary Computing, 2012
17. A Almansa, A Desolneux, S Vamech, Vanishing point detection without any a priori information. IEEE Trans. on Pattern Anal. Mach. Intell. **25**, 502–507 (2003). doi:10.1109/TPAMI.2003.1190575
18. C Stauffer, WEL Grimson, Adaptive background mixture models for real-time tracking, IEEE Conference on Computer Vision & Pattern Recognition. Colorado, USA. pp. 246–252. June 1999
19. A Saxena, J Schulte, AY Ng, Depth estimation using monocular and stereo cues. Int. Joint Conf. Artif. Intell. 2197–2203 (2007)
20. V Cantoni, L Lombardi, M Porta, N Sicari, Vanishing point detection: representation analysis and new approaches, Dip. di Informatica e Sistemistica – Università di Pavia IEEE 2001
21. J Zhong, S Sclaroff, Segmenting foreground objects from a dynamic textured background via a robust Kalman filter. Ninth IEEE Int. Conf. Comp. Vis. **2**, 44–50 (2003)
22. A Monnet, A Mittal, N Paragios, V Ramesh, Background modeling and subtraction of dynamic scenes. Ninth IEEE Int. Conf. Comp. Vis. **2**, 1305–1312 (2003)
23. DS Lee, Effective Gaussian mixture learning for video background subtraction. IEEE Trans. Pattern Anal. Mach. Intell. **27**, 827–832 (2005)
24. TH Tsai, WT Sheu, CY Lin, Foreground object detection based on multi-model background maintenance, in *IEEE International Symposium on Multimedia Workshops*, 2007, pp. 151–159
25. YC Lin, SC Tai, Fast full-search block-matching algorithm for motion-compensated video compression. Proc. 13th Int. Conf. Pattern Recog. **3**, 914–918 (1996)
26. SL Kilthau, MS Drew, T Moller, Full search content independent block matching based on the fast Fourier transform. Int. Conf. Image Process. **1**, 669–672 (2002)
27. C Yan, Y Zhang, J Xu, F Dai, J Zhang, Q Dai, F Wu, Efficient parallel framework for HEVC motion estimation on many-core processors. IEEE Trans. Circuit Syst. Video Technol. **24**(12), 2077–2089 (2014)
28. C Yan, Y Zhang, J Xu, F Dai, L Li, Q Dai, F Wu, A Highly Parallel Framework for HEVC Coding Unit Partitioning Tree Decision on Many-core Processors. IEEE Sign. Process. Lett. **21**, 573–576 (2014)
29. DDD. http://www.dddgroupplc.com/.