Virtual View Synthesis for Free Viewpoint Video and Multiview Video Compression using Gaussian Mixture Modelling

D. M. Motiur Rahaman, *Student Member, IEEE*, and Manoranjan Paul, *Senior Member, IEEE*

***Abstract*— High quality virtual views need to be synthesized from adjacent available views for *free viewpoint video* and *multi-view video coding* (MVC) to provide users with a more realistic 3D viewing experience of a scene. View synthesis techniques suffer from poor rendering quality due to holes created by occlusion and rounding integer error through warping. To remove the holes in the virtual view, the existing techniques use spatial and temporal correlation in intra/inter-view images and depth maps. However, they still suffer quality degradation in the boundary region of foreground and background areas due to the low spatial correlation in texture images and low correspondence in inter-view depth maps. To overcome the above-mentioned limitations, we use a number of models in the *Gaussian mixture modeling* (GMM) to separate background and foreground pixels in our proposed technique. Here, the missing pixels introduced from the warping process are recovered by the adaptive weighted average of the pixel intensities from the corresponding GMM model(s) and warped image. The weights vary with time to accommodate the changes due to a dynamic background and the motions of the moving objects for view synthesis. We also introduce an adaptive strategy to reset the GMM modeling if the contributions of the pixel intensities drop significantly. Our experimental results indicate that the proposed approach provides 5.40–6.60-dB PSNR improvement compared with the relevant methods. To verify the effectiveness of the proposed view synthesis technique, we use it as an extra reference frame in the motion estimation for MVC. The experimental results confirm that the proposed view synthesis is able to improve PSNR by 3.15–5.13 dB compared with the conventional three reference frames.**

***Index Terms*— View synthesis, free viewpoint video, depth image based rendering, multiview video compression, *moving picture experts group* (MPEG), *international telecommunication union* (ITU).**

I. Introduction

FVV has attracted considerable attention in recent years as it provides freedom to users to observe a scene from different angles or viewpoints [1]–[3]. A large number of views with a small baseline are required to facilitate this luxury, which increases transmission bandwidth and storage data significantly. *Depth image based rendering* (DIBR) is a practical way to reduce storage and transmission bandwidth for multiview videos from color textures and their corresponding depth maps [1], [2]. However, in the DIBR technique, certain portions of the virtual view are not visible due to being blocked by the front objects, which are termed as occlusions. Occlusions may create holes in the synthesized video [3]–[6]. Moreover, warping process from different views cause another source of error due to rounding of the pixel position coordinates.

Generally, there are two types of methods to fill missing pixels or holes, which are through spatial correlation and temporal correlation. In the spatial domain, spatial correlation of the video is exploited to fill the missing pixels. To reduce number of holes, view blending approaches can be used since two adjacent cameras can cover a relatively wider viewing angle [6]. In this technique, adjacent warped views are combined into a single view, which can reduce the holes. However, only a small number of views are transmitted due to the bandwidth constraint. Therefore, the rendered view could be missing some pixel information [7]–[10]. Inpainting is a popular technique to recover missing pixels by exploiting spatial correlation without introducing significant blur artifacts. The Inpainting technique used in [10] and [11] shows that after computing the priority of holes' boundary pixels, the most relevant patch is copied from the source patch. However, this process can deteriorate the quality of the view synthesis by being unable to differentiate foreground and background pixels properly. This is due to the low spatial correlation in the perimeter between foreground and background pixels [2], [5], [6], [10], [13]. In [14], blocks with missing pixels were sorted out in terms of decreasing difficulty for inpainting. In this technique, explicit instructions called *auxiliary information* (AI) of the most difficult blocks is transmitted to guide the decoder in the reconstruction process. The decoder can independently fill up missing pixels in the blocks that are easy to inpaint via a template-matching algorithm. In [15], depth information was used for priority computation and patch distance calculation of the algorithm in [11]. In this technique, the patch whose depth variance is low was given higher priority. However, this may produce distorted synthesized results around the foreground object boundaries when the boundaries of the objects in the depth map are mismatched

Manuscript received November 21, 2016; revised May 30, 2017, August 13, 2017, and October 3, 2017; accepted October 28, 2017. Date of publication November 13, 2017; date of current version December 5, 2017. The work was supported by the Australian Research Council under Grant DP130103670. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Rahul Vanam. *(Corresponding author: Manoranjan Paul.)*

The authors are with the School of Computing and Mathematics, Charles Sturt University, Bathurst, NSW 2795, Australia (e-mail: drahaman@csu.edu.au; mpaul@csu.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2772858

with that of the color images. Inverse mapping is another popular technique for hole-filling. This technique re-maps the missing pixel locations in the original view based on the column-shifts of the neighbourhood. In this way, holes can be mapped backwards to one of the original views to identify the missing pixel values [7], [12]. As this technique also exploits spatial correlation for the column part, it also suffers the hole filling problem in the foreground-background boundary areas.

Another method is to use temporal correlation to fill missing pixels of the view synthesis. This method is popularly known as background update technique and are based on the assumption that an occluded background in one frame may become visible in other frames when the foreground object(s) move away. The techniques used in [13] and [16] generated a static background frame by exploiting temporal correlation and then removing any foreground object with conventional inpainting and clustering techniques depending on the depth map. The experimental results revealed that these techniques improved the quality of the view synthesis significantly compared to other techniques including inpainting techniques [2]. However, this technique suffers quality degradation, due to the dependency on inpainting, warping of a background image and clustering methods. Inaccuracy of any these steps deteriorates the quality of the view synthesis. Moreover, an imperfect depth map may lead some artifacts from the background to the foreground. In addition, if the background frame is generated from the GMM model which is sorted by ratio of weight and standard deviation, it does not represent the recent changes of the pixel. This causes a poor background frame in view synthesis [2], [6], [17].

Unlike the previous GMM-based techniques [13], [16], our proposed technique uses a number of models in the GMM to separate background and foreground pixels and modifies the pixel intensities accordingly. We use an adaptive weighted average to generate the pixel intensities that is used to overcome the error introduced in the warping process. We also use an adaptive reset mechanism to keep the relevancy of the GMM modelling system.

View synthesis techniques are recognized as a promising tool for rendering views from *multiview video plus depth* (MVD) to support advanced 3D video coding [1], [18]. Recently, *international organization for standardization* (ISO), *moving picture expert groups* (MPEG) and *international telecommunication union* (ITU) video coding experts group have jointly developed efficient video coding tools such as 3D-HEVC [1], [3], [19], [20]. The main focus of the technique in [21] is to integrate a synthesized or disparity-adjusted view into the block-based *rate-distortion* (RD) optimization framework to improve prediction in MVD. For this, they generated a virtual view and introduced a skip and direct mode using the synthesized view. However, they did not include any explicit hole filling techniques to improve the quality of the synthesized view to address occlusion and error due to rounding integer problems. Therefore, the view synthesis prediction in [21] does not provide significant compression ratio improvement compared to 3D-HEVC.

3D-HEVC provides the best compression ratio for MVD data by exploiting the *view synthesis optimization* (VSO) coding tool [3]. A VSO technique for the exact view synthesis distortion calculation was proposed by employing a measure called *synthesis view distortion change* (SVDC). The view rendering was performed iteratively in the encoding process to compute the RD performance. This technique achieves high compression efficiency, but it inflicts heavy computation burden to the encoder. In another method [22], view synthesis distortion and depth distortion models without view rendering were proposed to reduce computational complexity. However, the accuracy was found to be lower than expected [22]. In [23], view synthesis distortion estimation for AVC and HEVC compatible 3-DV coding technique was proposed, where the view synthesis distortion function consistently achieved positive coding gains. To enable autostereoscopy additional views, the receiver generates the current frames from already encoded adjacent frames and the previous frame of the current view [24]. Moreover, DIBR techniques provide an extra reference frame by exploiting disparity among adjacent views. Due to the high similarity of the proposed view synthesis with the current view, this technique provides better prediction compared to the conventional three reference (i.e. two frames from adjacent views and the previous frame of the current view) technique. To verify the effectiveness of the proposed view synthesis, we use the generated frame as an additional reference frame in the motion estimation for MVC. The experimental results confirm that the proposed view synthesis is able to improve PSNR significantly compared to the conventional three reference frames. As we do not need any motion estimation for the virtual view, the computational time of using four reference frames is comparable to the three reference frames. We also use two reference frames which are generated using the proposed view synthesis frame and the previous frame. The results show that we can improve the PSNR compared to the three reference technique in multiview compression. This proposed two reference technique also reduces computational time significantly.

The preliminary concept of the view synthesis technique is published in [2]. The new contributions in this paper are (i) adaptive weighting, (ii) adaptive reset strategy for pixel modelling, (iii) a new way to generate pixel intensity of the virtual view, (iv) view synthesis using synthesized images, (v) introducing four reference and two reference techniques instead of the standard three reference MVC.

The rest of this paper is organized as follows: Section II describes the proposed view synthesis approach with adaptive weight hole filling technique, Section III focuses on view synthesis for MVC, while Section IV presents experimental results. Finally, the conclusions are given in Section V.

II. Proposed View Synthesis Technique

In few of the previous researches [13], [16], GMM technique is used for view synthesis using background frame. However, in our proposed technique, the number of models in the GMM is used to separate background/foreground pixels and modify pixel intensities. This is done using the corresponding model pixel intensities not only for the background model but also other models available in the GMM. The missing pixels of the background are recovered using adaptive weighted average of the pixel intensities from the model(s) and the warped image, to overcome the error introduced in the warping process. In this technique, the inherent characteristics of Gaussian mathematical models are capitalized to recover occluded areas. It is true that the GMM technique is more effective for static background scenarios, however, it is also useful to address pixel intensity problem for the event of occlusion. Moreover, with minor changes to the technique, it can handle dynamic background scenarios. To handle more dynamic background and foreground scenarios, we have used an adaptive reset mechanism in the proposed method when the current models lose their relevancy. Moreover, unlike in general scenarios, static cameras are used in the free viewpoint and multiview video scenarios.

In the GMM technique, if a pixel in a certain position experiences similar intensities over a period of time, it represents only one model which indicates that the pixel is a background. On the other hand, if a pixel in a certain position experiences different pixel intensities, then it is represented with multiple Gaussian models. This indicates that the pixel has both background and foreground at different times. Therefore, the hypothesis is that the number of GMM models would be a good indicator to identify background/foreground pixels. In this technique, the GMM is applied on the interpolated view instead of the adjacent view assuming that synthesized previous images of the interpolated view are already available [2]. This technique provides a better pixel correspondence, which leads to better quality compared to both inpainting and background update methods. In [2], if a pixel in a certain position experiences foreground once, but is at the background in other moments, it is considered foreground throughout the technique. However in reality, after experiencing foreground intensity a pixel can also experience background intensity again. Based on this hypothesis, we find appropriate background and foreground pixels for filling missing pixel intensities of the virtual view.

Moreover, the setting weight is used to selected a portion of the bended image and GMM model, this is crucial as the PSNR of the view synthesis may vary from 1.0∼6.0dB by using different weights to balance the contributions between warping image and the learned foreground model. In this paper, we propose an adapting weighting technique to fill up missing pixels of the view synthesis. In the experiment, we have observed that if a video has more moving regions, the tendency is that it has more pixels which use two or more Gaussian models. In this situation, the relatively large contribution from warped image provides a better quality of the view synthesis. It is due to the less relevancy of the learned foreground with the view synthesis for the rapid changes of foreground within a short period of time. In this paper, we first established a relationship between the weight and the percentage of multiple Gaussian models using a certain number of videos. Then, we apply the relationship in the generation of view synthesis. The experimental results show that the proposed technique does not sacrifice any significant quality degradation compared to the maximum achievable quality through setting the weights.

In the proposed technique, n-th texture images from two adjacent views are warped into a virtual position by using their corresponding depth maps and camera parameters to generate the n-th image of the intermediate view. However, warped images contain holes due to the occlusion and rounding integer error. Two warped images are blended to reduce these missing pixels to make a bended image. This procedure reduces the number of holes but does not help to recover all missing pixel intensities specially occluded regions. To recover these missing pixels, we use GMM technique to model each pixel with available previous frames of the virtual view as shown in Fig. 1. In our experiment, we assume that we have already 1 to $(n-1)$-th frames for the GMM when we generate n-th frame of a virtual view. In this technique, parameter i is used to reset the pixel modelling after a certain interval, where $i = 2, 3, 4 \ldots n$. The resetting of the modelling depends on weighting factor (for details, see Section II (D)). Initially, we use the original frame for GMM, next, we also use the synthesized frame. Then, based on the number of GMM models, each pixel is classified as a foreground or background pixel. From the background models, we generate a *most common frame in scene* (McFIS) to recover the missing pixels [17]. After that, the missing pixel intensities of the background and foreground areas are filled from the adaptive weighted intensities between the blended image and the learned background and foreground model(s) of the GMM. The subsequent section describes interpolating virtual view, GMM technique, adaptive hole filling technique, and choosing the value of weighting factor.

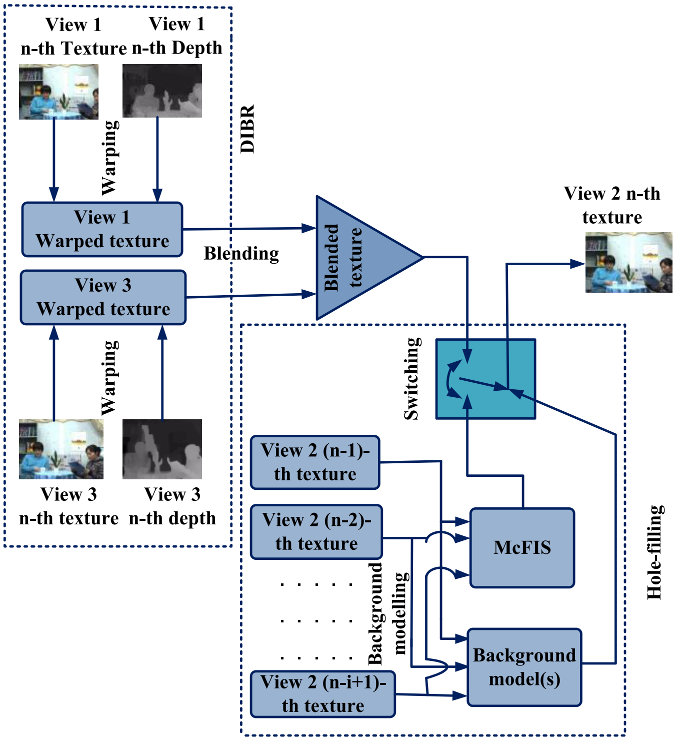


Fig. 1. Proposed view synthesis technique.

A. Interpolating Virtual View

In our experiment, we assume that the sender usually transmits two texture images and their correspondence depth maps of a same scene is captured by two cameras at the same instant. Generally, depth maps represent the distance of objects from the camera which is quantized into 256 different values where 0 and 255 represent the farthest and nearest distance respectively. The true depth values Z are converted from the encoded depth map Ω by using (1) [2], [7]:

$$z = \frac{Z_{near} Z_{far}}{\left(\frac{\Omega}{255}\right)\left(Z_{near} - Z_{far}\right) + Z_{near}}. \tag{1}$$

where Z_{far} and Z_{near} are the farthest and nearest depth in a scene respectively.

Then the disparity (d) between the reference view (adjacent view) and the virtual view is determined from the camera parameters which are camera focal length (f) and baseline (l) by using (2):

$$d = \frac{fl}{Z}. \tag{2}$$

After that, texture images are aligned in the virtual position based on the calculated disparity values [7], [25]. However, this aligned texture contains many holes due to rounding integer error and the occlusion problem. Warped images (Γ_1' and Γ_3') are blended based on four conditions to minimize the holes problems as follows:

Case I: If there are no holes in the warped texture Γ_1' and warped texture Γ_3', we take the average of the corresponding pixels.

Case II: If there are no holes in the warped texture Γ_1', but there are holes present in the warped texture Γ_3', we take the pixel intensity from the warped texture Γ_1'.

Case III: If there are holes present in the warped texture Γ_1', but there are no holes in the warped texture Γ_3', we take the pixel intensity from the warped texture Γ_3'.

Case IV: If there are holes present in both warped textures, we consider the pixel intensity is equal to zero.

This procedure reduces the number of holes but does not help to recover all missing pixel intensities. To recover the missing pixels, we model each pixel by using the GMM technique with previously generated images in the virtual view.

B. GMM Technique

The GMM technique is usually used for separating background and foreground pixels (pixel level) from a dynamic environment, where each pixel is modeled independently by a mixture of K-th Gaussian distributions (usual setting $K = 3$) [27], [28]. In our proposed technique, we assume that at time t, the value of k-th Gaussian intensity $= \eta_{k,t}$, mean $= \mu_{k,t}$, variance $= \sigma_{k,t}^2$, and weight in the mixture $= \omega_{k,t}$, so that $\sum_{k=1}^{K} \omega_{k,t} = 1$. We set the initial parameters (from [27], [29]) as follows: standard deviation $(\sigma_k) = 2.5$, weight $(\omega_k) = 0.001$ and learning rate, $\alpha = 0.1$. A learning parameter $0 < \alpha < 1$ is used for balancing the contribution between the present and previous values of aforementioned parameters.

After setting the initial parameters, the current pixels are used to match with k-th Gaussian for every new observation if the condition $\left|X_t - \mu_{k,t}\right| \leq 2.5\sigma_{k,t}$ is satisfied against existing models, where X_t is the new pixel intensity at time t. If a model matches, the Gaussian model will be updated as follows:

$$\mu_{k,t} \leftarrow (1-\alpha)\mu_{k,t-1} + \alpha X_t; \tag{3}$$

$$\sigma_{k,t}^2 \leftarrow (1-\alpha)\sigma_{k,t-1}^2 + \alpha(X_t - \mu_{k,t})^T (X_t - \mu_{k,t}); \tag{4}$$

$$\omega_{k,t} \leftarrow (1-\alpha)\omega_{k,t-1} + \alpha, \tag{5}$$

and the weights of other Gaussians models are updated as

$$\omega_{k,t} \leftarrow (1-\alpha)\omega_{k,t-1}. \tag{6}$$

Then, the value of weights are normalized among all models in such a way that $\sum_{k=1}^{K} \omega_{k,t} = 1$. Conversely, if a model fails to match, then a new model is introduced with initial parameter values. If it has already crossed the maximum allowable number of models, based on the value of weight/standard deviation, a new model substitutes the existing model. If a pixel intensity of a color (c) satisfies a model (k), we store the pixel intensity as *recent value* ($\mathrm{B}_{k,t}^c$) of the corresponding model and color. After that, we use this value to recover missing pixel values.

C. Hole Filling

If a pixel experiences only one model over time in different frames, it represents a static background pixel. Conversely, if a pixel experiences more than one model, it represents foreground or background pixel, where the highest value of weight/standard deviation represents the most stable background [2]. As the GMM has inherent capacity to capture background and foreground pixel intensities, missing pixel intensities of an occluded area are successfully recovered by exploiting temporal correlation. In the proposed technique, if a pixel is considered a static background pixel, the pixel intensity of the synthesized final image (Ψ_t^c) is taken from the *recent value* ($\mathrm{B}_{k,t}^c$) of the model and warped image. However, a video with larger moving objects and faster motion, changes the video content frequently, as a result, the models lose relevancy with the past frames. In this scenario, learned foreground using GMM does not provide suitable pixel intensity for a virtual view. Therefore, we need to reset the models after a certain interval, otherwise, error propagates through the whole system. On the other hand, the pixel intensities of the synthesized final image is taken as a weighted average from the blended image and the *recent value* of the model, which provides the lowest value in terms of weight/standard deviation. The details of the interpolated image recovering technique using GMM is described below:

Case 1: If a pixel experiences only one model over the whole duration for a given colour, we store the recent value ($\mathrm{B}_{k,t}^c$) of the colour for the final image synthesis by using

$$\Psi_t^c = (\xi - 0.5842)\Phi_t^c + (1.0 + 0.5842 - \xi)\mathrm{B}_{k,t}^c. \tag{7}$$

where ξ is the weighting factor (see detail calculation of constants used in (7) in the subsection D of section II) and Φ_t^c is the outcome of inverse mapping.

Case 2: If a pixel experiences more than one model for a given colour over the duration, whether it would be the foreground or background pixel, initially, we use the inverse mapping technique [7] to fill the holes. Then, we find the smallest difference between the pixel intensities of Φ_t^c and the *recent values* $\mathrm{B}_{1,t}^c$, $\mathrm{B}_{2,t}^c$ and $\mathrm{B}_{3,t}^c$ as follows:

$$\begin{aligned}\Delta_1 &= \left|\Phi_t^c - B_{1,t}^c\right| \\ \Delta_2 &= \left|\Phi_t^c - B_{2,t}^c\right| \\ \Delta_3 &= \left|\Phi_t^c - B_{3,t}^c\right| \\ \Delta &= \min(\Delta_1, \Delta_2, \Delta_3) \end{aligned} \tag{8}$$

If $\Delta = \Delta_1$, the pixel represents the background at that moment for a given colour and we store the recent value $\mathrm{B}_{1,t}^c$ of the colour for the final synthesized image by using (7).

If $\Delta = \Delta_2$ or $\Delta = \Delta_3$ the pixel represents foreground at that moment, therefore, we choose a weight factor (ξ) for selecting a portion of the Φ_t^c and a portion of the *recent value* of the second or third model ($\mathrm{B}_{2,t}^c$ or $\mathrm{B}_{3,t}^c$) as follows:

$$\Psi_t^c = \xi\Phi_t^c + (1-\xi)\mathrm{B}_{k,t}^c. \tag{9}$$

where the value of k is either 2 or 3.

D. Adaptive Weighting Factor

In the proposed technique, we have observed that different values of weighting factor ξ provide different qualities of virtual view (see Fig. 6). Thus, it is essential to determine the value of ξ in different frames and videos. To determine the value of ξ, we have tried to learn the factor which influences the value so that it provides a better virtual view. Our theory is that, if a video has larger foreground areas with fast motion, the video should have larger number of pixels with multiple models in GMM. In this case, the video should show a tendency to take more pixel intensities from the warped image compared to the background image as the background image loses its relevance more frequently over the time. Thus, the value of ξ would be proportionate with the number of multiple models in GMM. Through experiments, it has been observed that there is a positive relationship between the value of ξ and number of pixels with multiple models of a frame. In this scenario, learned foreground using GMM does not provide suitable pixel intensities for a virtual view, thus the value of ξ should be higher for those cases as the warped image has more contribution compared to the learned foreground. We derive a relationship between multiple models and a weighting factor (ξ) for a number of videos. Then, we use this relationship in each frame of the video to adaptively set the value of the weighting factor. The weighting factor is formulated as a ratio of the number of pixels with multiple models and the total number of pixels in a frame in GMM, which is given below:

$$\begin{aligned}\xi &= f(A) \\ &= \frac{A_2 + A_3}{A_1 + A_2 + A_3} \times 100 \\ &= A_{2,3}\end{aligned} \tag{10}$$

where A_1, A_2 and A_3 are the number of pixels with one model, two models, three models respectively and $A_{2,3}$ is the percentage of the number of pixels which possesses multiple models. From (10), we fit a third order polynomial $(0.0004(\mathrm{A}_{2,3})^3 - 0.0116(\mathrm{A}_{2,3})^2 + 0.1214\mathrm{A}_{2,3} + 0.5842)$ to derive the relationship that is shown in Fig. 2. Here to be noted that we have calculated the weighting factor in $(n-1)$-th frame and used it to generate n-th virtual frame.

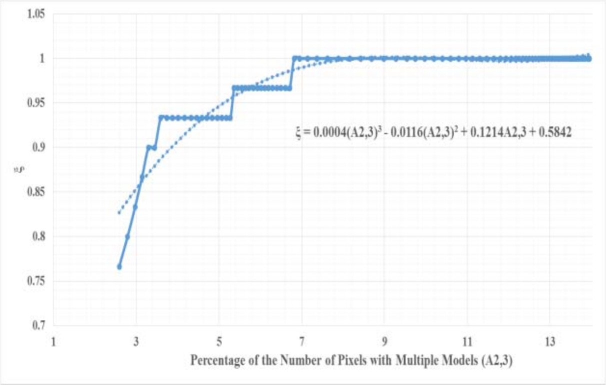


Fig. 2. Trend of the weighting factor (ξ).

The main idea of the adaptive weighting factor determination is that if a video has larger moving objects, the large contribution comes from warped images ensure better view synthesis. Moreover, it helps to reset the pixel modelling after certain intervals depending on the number of multiple models. When the number of multiple models increases, the contribution of learned background/foreground reduces to ensure better view synthesis. In our experiment, when the value of ξ is 0.9 or higher, we reset the modelling. Fig. 2 shows that the value of ξ is very close to 1 when the number of multiple models is close to 13% or more. This means that there is little or no contribution of learned foreground to form a virtual view. However, the learned background of GMM still has some contribution when forming the virtual frame in static and uncovered background areas. When we compare the adaptive weighting factor to generate a virtual view, we do not sacrifice any significant quality degradation compared to the maximum achievable quality by setting the weights from 0 to 1 (see Fig. 7).

III. View Synthesis for MVC

Adjacent views of multiview video sequences are captured by multiple cameras with slightly different angles. Therefore, there are disparities among the different views. Here, to predict the co-located pixels/blocks at different instances of the same views, motion estimation technique is used. However, finding co-located pixels/blocks on different frames by using motion estimation and disparity estimation is time consuming [30], [31]. Therefore, a reduction of computation for searching motion parameters such as motion vector is an important aspect of the current research [32]–[34]. The best policy to reduce this computational time is by reducing the number of reference views. Traditionally, three reference technique uses already encoded frames of adjacent views (reference frame 1 and 2 in Fig. 3) and the previous frame of the current view (reference frame 3 in Fig. 3) to encode each frame of dependent view [3], [24]. In this technique,

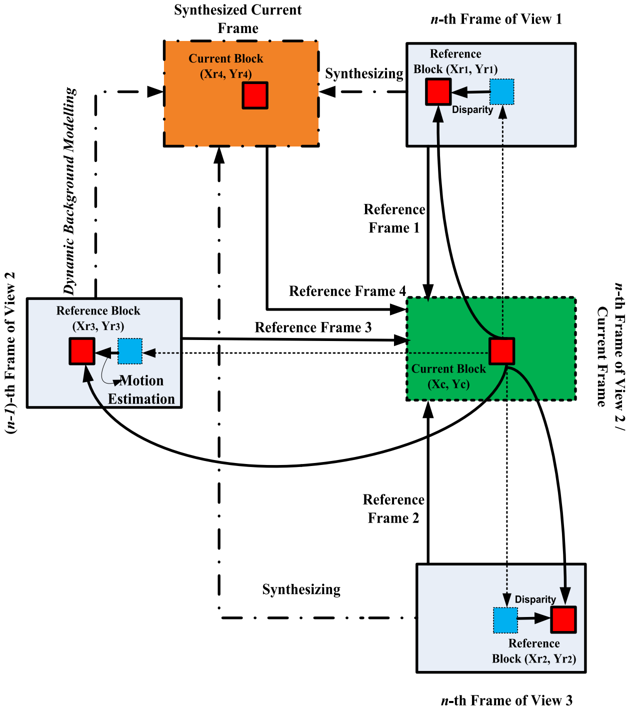


Fig. 3. Proposed MVC coding technique by using four reference such as (n-1)-th frame of view 2, n-th frame of view 1, n-th frame of view 3 and the virtual frame generated by the proposed technique.

a disparity d is used to find a *current block* (X_c, Y_c) on the adjacent reference views $((X_{r1}, Y_{r1})$ and $(X_{r2}, Y_{r2}))$ where $X_{r1} = X_c \mp d$ and $X_{r2} = X_c \pm d$. This method only considers the horizontal component as multiview video sequences are rectified [3]. Furthermore, motion vectors are predicted to find a current block on the previous frame of the current view (X_{r3}, Y_{r3}) [29], [35]. Instead of typical approaches, we use the proposed view synthesis technique to generate a synthesized current frame, which is used as the fourth reference frame. This synthesized frame is almost similar in terms of the object position and its motion to the expected frame. Therefore, we have four candidates (references) for choosing each block to encode the current frame of the middle view (view 2) as shown in Fig. 3. As the fourth reference frame has more similar content with the current frame compared to the other three reference frames, it is expected that encoding the current frame using four reference frames provides better quality. Moreover, using the four reference frames does not require significant extra computational time compared to three reference frames as it does not require any disparity or motion estimation.

To see the effectiveness of the proposed virtual frame, we also consider two reference (reference three and reference four) as shown in Fig. 3, this technique also provides better prediction compared to the traditional approaches. As can be seen, two reference technique provides better computational time compared to three reference frame technique.

IV. Experimental Results

In our experiment, PSNR is used to measure the squared intensity differences of synthesized and original image pixels.

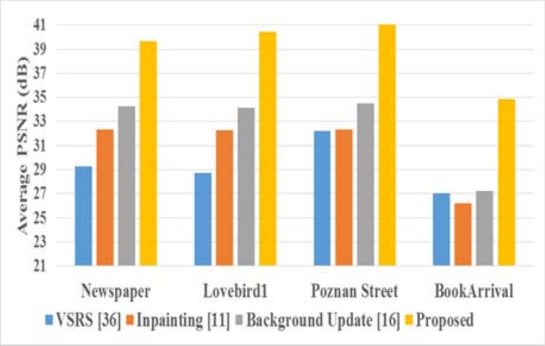


Fig. 4. Average PSNR comparison for 100 frames.

TABLE I
Test Sequences, Synthesized Viewpoints and Baseline

Sequences	Input Reference Viewpoints	Target Viewpoint	Baseline
Newspaper	6,2	4	185.37
Lovebird1	8,4	6	148.12
Poznan Street	5,3	4	3.19
Book Arrival	10,6	8	2.31

Then, based on average PSNR performance, we compare the outcome of the proposed method with the state-of-the-art methods, namely, *View Synthesis Reference Software* (VSRS) [36], inpainting [11], and background update technique [16]. Four standard multiview video sequences are selected for testing the performance of our proposed technique. Table 1 lists the input reference viewpoints, the virtual viewpoints and the baseline of the four video sequences used. We use the same warping and blending techniques for 100 frames using adjacent views, then we apply VSRS, inpainting, background update and the proposed method for refining the blended image. Fig. 4 shows that the proposed technique provides better performance compared to the existing hole filling techniques for all video sequences. The improvement range varies from 7.85dB to 11.69dB (with average improvement 9.72dB) for VSRS, 7.32dB to 8.85dB (with average improvement 8.25dB) for inpainting and 5.40dB to 7.65dB (with average improvement 6.51dB) for background update technique respectively. Furthermore, when we compare the outcome of our preliminary paper [2] with the proposed technique (as shown in Fig. 5) we find that the proposed technique outperforms the preliminary technique for all video sequences. In [2], only a single view was taken for warping, whereas the proposed technique uses two adjacent views for warping. Moreover, he proposed method identifies both foreground and background pixel intensities to refine the virtual view when multiple models are used to model pixel intensities. The model which provides the least difference in pixel intensities between the blended image and the different models in GMM for a given moment, represents either foreground or background pixel intensity. However, in the previous method [2], it was considered as a foreground pixel intensity throughout. That is why PSNR of the virtual view using the proposed method increases with the number of frames compared to the technique in [2].

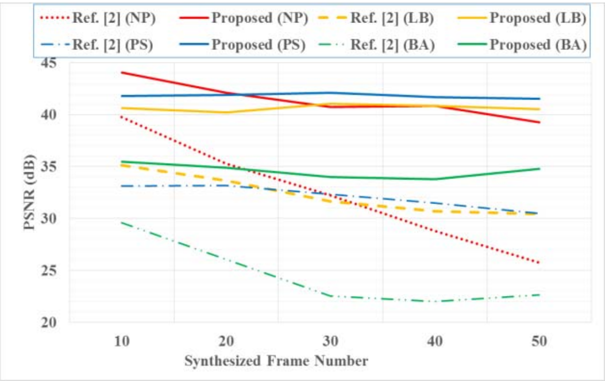


Fig. 5. PSNR Comparison of the technique in [2] against the proposed technique for *Newspaper* (NP), *Lovebird1* (LB), *Poznan Street* (PS) and *Book Arrival* (BA) video sequences.

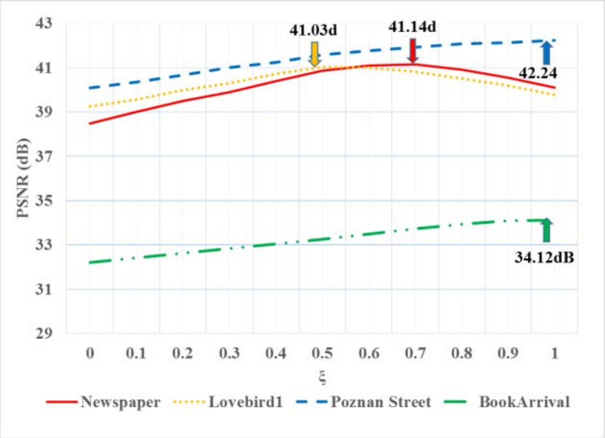


Fig. 6. Weighting factor (ξ) vs PSNR (dB).

We analyzed PSNR against values of ξ ranging from 0 to 1. In our proposed approach, both GMM models and blended images contributes in reconstructing of the final synthesized images. This can be seen in 30th frame of each of the four video sequences in Fig. 6. Here to be noted that if we get the maximum PSNR value for a given image where the value of ξ is 0.6, it means that 60% and 40% of the foreground pixel intensities are taken from blended image and learned foreground respectively. We observed that, while a fixed threshold may work for some frames, not all frames have the same threshold. Thus, it is crucial to use adaptive threshold rather than a fixed threshold.

We also compared the maximum PSNR and adaptive PSNR against predicted frames for *Newspaper*, *Lovebird1*, *Poznan Street* (up to 100 frames each) and *Book Arrival* (full sequence), which is shown in Fig. 7. From this figure it can be seen that *Newspaper*, *Lovebird1*, *Poznan Street* and *Book Arrival* video sequences sacrifice 0.10dB, 0.08dB, 0.16dB and 0.23dB PSNR on average. The slope of this curve is controllable by changing the values of ξ. If the value of ξ is greater than or equal to 0.9, we reset the pixel modelling in this experiment, as this causes the contributions from the GMM models to reduce (0.1 or less). In our experiment, the pixel modelling is reset 10, 5, 9 and 14 times for *Newspaper, Lovebird1, Poznan Street and Book Arrival* video sequences respectively. Usually, due to the content of the video, the PSNR of the view synthesis occasionally rises or falls over time. Therefore, the proposed reset strategy is devised to handle this trend of changing PSNR over time by reducing the contributions of the pixel intensities of the GMM model.

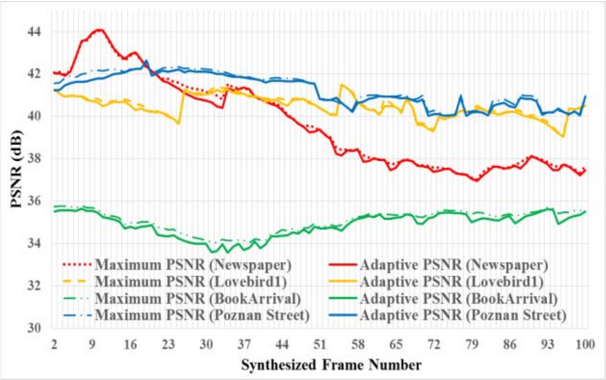


Fig. 7. Maximum PSNR (dB) vs adaptive PSNR (dB) for 100 frames for four video sequences by learning original frames.

Fig. 8 illustrates the subjective quality for the *Newspaper* video sequence. Fig. 8 (a) shows the original image (10th original frame) of the virtual view and Fig. 8 (b) and (c) shows green rectangular boxes that are used to mark the cropped and zoomed portion. Fig. 8 (d), (g), (j) and (m) shows the synthesized images by VSRS, inpainting, background update and the proposed technique respectively and Fig. 8 (e), (f), (h), (i), (k), (l), (n) and (o) shows the corresponding cropped and zoomed images. Fig. 8 demonstrates that the proposed technique is able to generate a better view synthesis compared to the other three methods mentioned.

To see the effectiveness of the proposed method, we have applied the proposed technique to the synthesized images generated by the inverse mapping technique. Fig. 9 shows that the proposed technique improves the quality of the synthesized view. This technique improves 0.57dB, 0.15dB, 0.27dB and 0.32dB PSNR on average for *Newspaper, Lovebird1, Poznan Street* and *Book Arrival* video sequences respectively compared to the inverse mapping technique. Fig. 10 demonstrates the subjective quality of the proposed technique when we learned the output of the inverse mapping technique [7]. It shows that the proposed technique provides better synthesized images. As the parameters of the proposed technique are not optimized for the synthesized views, it gives us moderate improvement. If better quality images for learning GMM were available, it would provide better synthesized images.

To understand the effectiveness of the proposed view synthesized technique in the moving background sequences, we have conducted experiments using *Balloons*, *Kendo*, *Poznan Hall2* and *Undo Dancer* video sequences with 50 frames each. Fig. 11 shows the performance of the proposed method compared to Inverse mapping method [7]. Fig. 11 also shows that the proposed method performs better for *Ballons* and *Kendo* video sequences but not for the other two sequences. This is because the first two video sequences

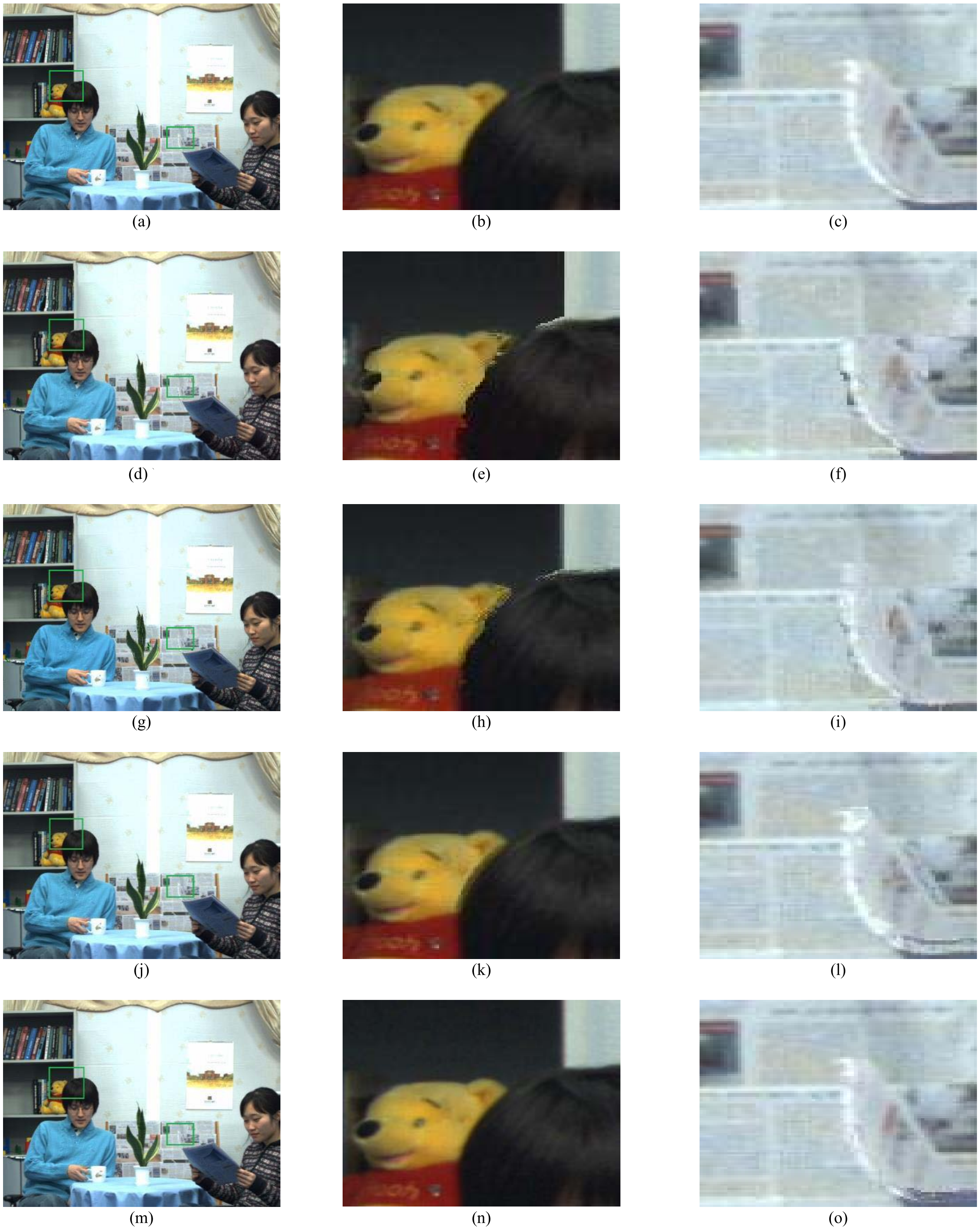


Fig. 8. (a) Original frame. (b) Cropped and zoomed image of the original frame. (c) Cropped and zoomed image of the original frame. (d) VSRS technique [36]. (e) Cropped and zoomed image of the VSRS technique [36]. (f) Cropped and zoomed image of the VSRS technique [36]. (g) Inpainting technique [11]. (h) Cropped and zoomed image of the inpainting technique [11]. (i) Cropped and zoomed image of the inpainting technique [11]. (j) Background Update technique [16]. (k) Cropped and zoomed image of the background update technique [16]. (l) Cropped and zoomed image of the background update technique [16]. (m) Proposed technique. (n) Cropped and zoomed image of the proposed technique. (o) Cropped and zoomed image of the proposed technique.

have relatively less moving background compared to other two video sequences. Thus, the proposed technique still provides better results if the moving background is not too fast.

To encode different resolutions and a wide range of video content for different views in 3D-HEVC, each frame is divided into a number of blocks with various sizes such as 8×8, 16×16, 32×32 and 64×64 pixels [1]and the search length become 8, 16, 32, 64 and 128 pixels. In our experiment, we have used two types of block sizes which are 32×32 pixel block (with 64 pixel search length) and 64×64 pixel block (with 64 pixel search length) to demonstrate the performance of the proposed four and two reference techniques compared to the existing three reference technique. Due to

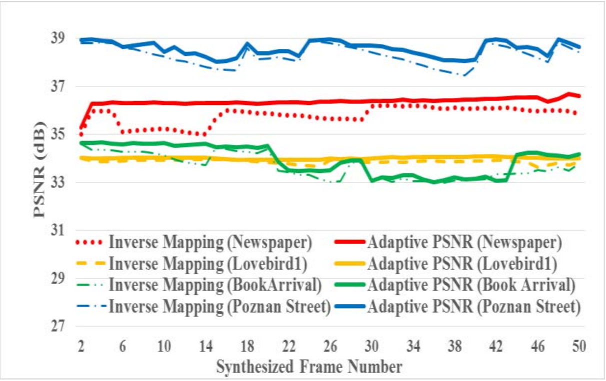


Fig. 9. Inverse mapping vs adaptive PSNR (dB) for four video sequences when learning the output of inverse mapping for 50 frames.

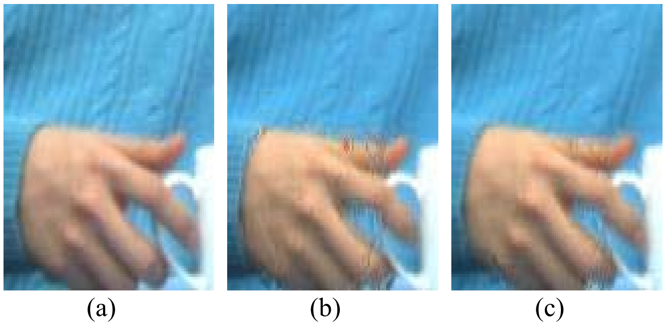


Fig. 10. (a) Original image, (b) output images of inverse mapping technique and (c) proposed image after learning inverse mapping output.

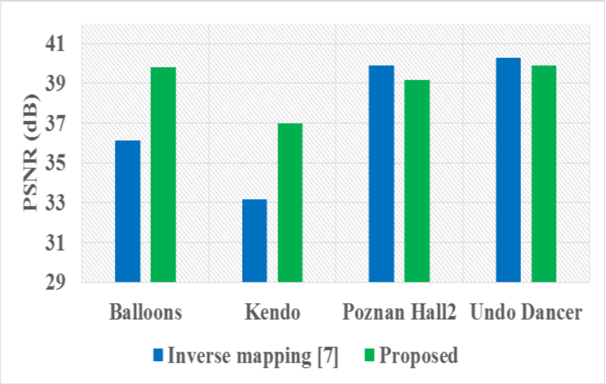
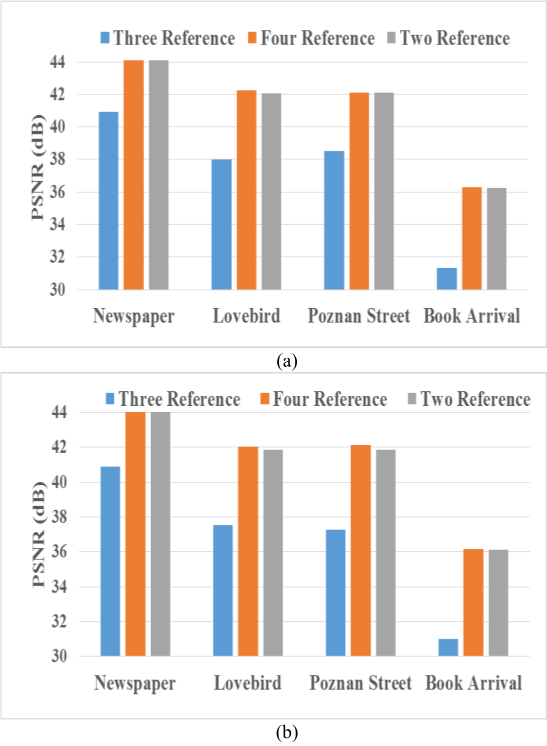


Fig. 11. Average PSNR comparison for moving background video sequences for 50 frames using the adaptive weighted view synthesized technique.

the better prediction of the synthesized view, the proposed technique provides better PSNR compared to the conventional approaches, which are shown in Fig. 12(a) for 32×32 pixel block and Fig. 12(b) for 64×64 pixel block. These figures show that the PSNR improvement in the two reference and four reference techniques are in the ranged of 3.18 to 4.95dB, when block size is 32×32 pixels. Similarly, when the block size is 64×64 pixels, the PSNR improvement in two reference and four reference technique are in the ranged of 3.15dB to 5.14dB. The four reference technique provides better PSNR compared to the two reference and three reference techniques.



(a)

(b)

Fig. 12. PSNR comparison for the proposed MVC techniques: (a) 32×32 pixels block size motion estimation and (b) 64×64 pixels block size motion estimation.

Moreover, for the performance verification of the proposed two reference and four reference techniques for static cameras (*Newspaper* and *Lovebird1*) and moving cameras (*Balloons* and *Undo Dancer*) video sequences, we have generated the RD performance curve using different QPs (22, 27, 32 and 37) in the scenario of MVC as shown in Fig. 13. We have used 3D-HEVC structure where the first view and third view are encoded using HEVC coding framework. The middle view is encoded using two adjacent inter-view images, the immediate previous intra-view image, and the synthesized image. All reference frames are generated from the reconstructed (decoded) reference frames, so that, both encoder and decoder have the same reference frames. We compared the effectiveness of the proposed method in terms of generating better synthesized view which is used as one of the reference frames for coding purposes. The experimental results illustrate that both the proposed techniques improve RD performance significantly in all the video sequences by improving the quality of the synthesized views.

Furthermore, the performances of the four reference and two reference techniques against the three reference technique are evaluated based on the *Bjøntegaard-Delta Bit Rate* (BD-BR) and *Bjøntegaard-Delta PSNR (*BD-PSNR) [37] in Table 2, where '+' and '−' sign indicate the increment and decrement respectively. For all eight different video sequences, the four reference and two reference techniques provide gains 1.07dB and 0.88dB BD-PSNR, while the BD-BR decreases by 29.68% and 26.06% on average respectively compared to the conventional three reference technique. The proposed method outperforms the three reference technique for all video

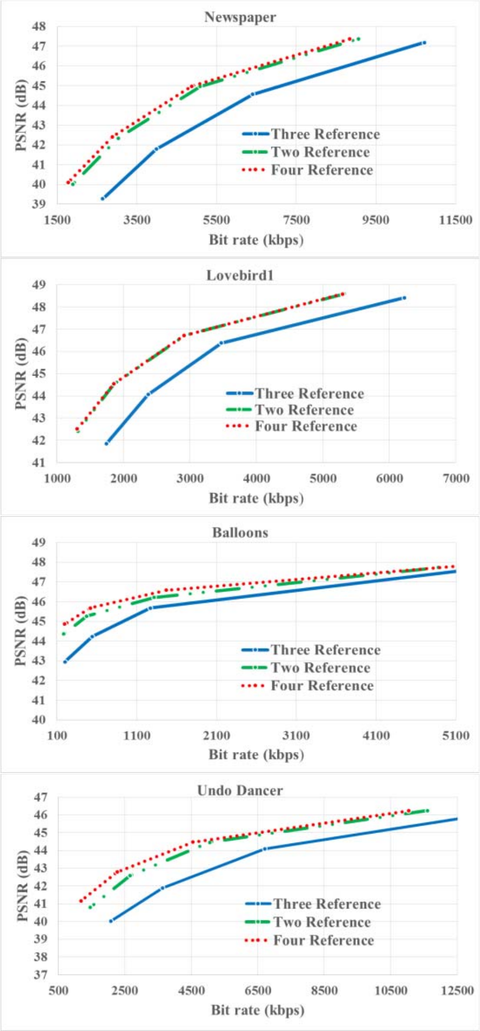


Fig. 13. Rate-distortion performance relationship using three reference, two reference and four reference techniques for four video sequences.

TABLE II

THE PERFORMANCE OF THE PROPOSED FOUR REFERENCE AND TWO REFERENCE METHODS AGAINST THE EXISTING THREE REFERENCE TECHNIQUE USING BD-BIT RATE AND BD-PSNR

Sequences	Four Reference		Two Reference	
	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)	BD-BR (%)
Newspaper	+1.92	-33.03	+1.67	-27.75
Lovebird1	+0.77	-23.35	+0.72	-22.74
Poznan Street	+0.48	-14.43	+0.43	-13.65
Book Arrival	+1.7	-39.12	+1.62	-38.53
Balloons	+0.75	-38.33	+0.47	-35.67
Kendo	+0.80	-23.88	+0.53	-16.88
Poznan Hall2	+0.80	-28.99	+0.6	-24.14
Undo Dancer	+1.32	-36.28	+0.6	-24.14
Average	**+1.07**	**-29.68**	**+0.88**	**-26.06**

sequences in terms of both improving the BD-PSNR and reducing the BD-BR.

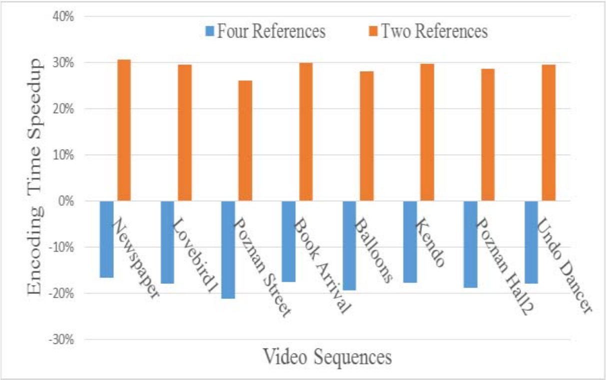


Fig. 14. Encoding time speedup for the proposed four and two reference techniques compared to the traditional three reference technique for MVC.

MVC leads to high computational complexity, which limits its application on low power consumption electronic devices such as smart phones [29]. Total encoding time heavily depends on motion and disparity estimation. Research shows that there are no significant time differences for estimating motion and disparity [30], [31]. Moreover, MVC exhaustively checks a number of inter/intra modes for a coding unit to select the best mode for encoding. This procedure increases complexity multiple times compared to the unimode technique [38]. Therefore, any technique which skips disparity estimation and/or motion estimation should reduce time complexity. The proposed two reference technique skips disparity estimation and improves PSNR compared to the three reference technique. Although the proposed technique needs extra computational time for synthesis virtual view, it reduces overall time complexity for MVC by 28.95% whereas the four reference technique requires an extra 18.42% time on average compared to the existing three reference technique (see Fig. 14). In both cases, the proposed technique outperforms the three reference technique in terms of image quality for a given bit rate. All experiments are conducted on a dedicated desktop machine DELL OPTIPLEX 9020 (with Intel core i5-4690 CPU @ 3.50 GHz, 8 GB RAM and 250 GB HDD) running 64-bit Windows 7 operating system. According to the rate-distortion performance (see Table 2 and Fig. 13), the proposed four reference technique outperforms the proposed two reference technique for all video sequences. Significant performance gains are observed for the video sequences with camera motions (*Undo Dancer*, *Kendo and Balloons*). However, the proposed four reference technique requires around 47% extra computational time compared to the two reference technique. Thus, our recommendation is to use the four reference technique for the scenarios where the video sequences has camera motions, but there is no concern of the computational time requirements, otherwise, two reference technique can be used.

V. CONCLUSION

In this paper, we presented a new view synthesis technique that exploits temporal correlation for hole filling. In our proposed technique, views are interpolated from adjacent texture images and their corresponding depth maps. Interpolated images contain many holes due to the occlusion and rounding integer problems. Usually, spatial and/or temporal correlation-based techniques (inpainting and background updates) are used to address these issues. However, these techniques suffer quality degradation due to the low spatial correlation in the boundary areas of the foreground and background pixels. Therefore, in our proposed technique, we use a number of models in GMM to separate background and foreground pixels. The missing pixels are recovered from the adaptive weighted average of the pixel intensities from the corresponding model(s) of the GMM and the warped images. Experimental results shows that the proposed technique provides 9.72dB, 8.25dB and 6.51dB PSNR improvement on average compared with the VSRS, inpainting and background update techniques respectively. To evaluate the effectiveness of the proposed, we used the view synthesis from the decoded frames as an extra reference frame for MVC. This improves the quality of the encoded frame by an average of 0.73dB compared to the standard techniques. Another version of the proposed technique provides 0.68dB image quality improvement with reduced computational time when compared to the existing MVC technique.

References

[1] G. Tech, Y. Chen, K. Müller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3D extensions of high efficiency video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 35–49, Jan. 2016.

[2] D. M. M. Rahaman and M. Paul, "Hole-filling for single-view plus-depth based rendering with temporal texture synthesis," in *Proc. IEEE Int. Workshop Multimedia Expo Workshops (ICMEW)*, Jul. 2016, pp. 1–6, doi: 10.1109/ICMEW.2016.7574740.

[3] K. Müller *et al.*, "3D high-efficiency video coding for multi-view video and depth data," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 3366–3378, Sep. 2013.

[4] F. Zou, D. Tian, A. Vetro, H. Sun, O. C. Au, and S. Shimizu, "View synthesis prediction in the 3-D video coding extensions of AVC and HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, pp. 1696–1708, Oct. 2014.

[5] G. Luo, Y. Zhu, Z. Li, and L. Zhang, "A hole filling approach based on background reconstruction for view synthesis in 3D video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1781–1789.

[6] D. M. M. Rahaman and M. Paul, "Free view-point video synthesis using Gaussian mixture modelling," in *Proc. IEEE Conf. Image Vis. Comput. New Zealand*, Nov. 2015, pp. 1–6.

[7] M. S. Farid, M. Lucenteforte, and M. Grangetto, "Depth image based rendering with inverse mapping," in *Proc. IEEE 15th Int. Workshop Multimedia Signal Process.*, Sep. 2013, pp. 135–140.

[8] Z.-W. Liu, P. An, S.-X. Liu, and Z.-Y. Zhang, "Arbitrary view generation based on DIBR," in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst.*, Nov. 2007, pp. 168–171.

[9] C.-M. Cheng, S.-J. Lin, S.-H. Lai, and J.-C. Yang, "Improved novel view synthesis from depth image with large baseline," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.

[10] A. Oliveira, G. Fickel, M. Walter, and C. Jung, "Selective hole-filling for depth-image based rendering," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2015, pp. 1186–1190.

[11] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[12] D.-H. Li, H.-M. Hanh, and Y.-L. Liu, "Virtual view synthesis using backward depth warping algorithm," in *Proc. Picture Coding Symp.*, Dec. 2013, pp. 205–208.

[13] C. Yao, Y. Zhao, and H. Bai, "View synthesis based on background update with Gaussian mixture model," in *Proc. Pacific-Rim Conf. Multimedia*, 2012, pp. 651–660.

[14] Y. Gao, G. Cheung, T. Maugey, P. Frossard, and J. Liang, "Encoder-driven inpainting strategy in multiview video compression," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 134–149, Jan. 2016.

[15] I. Daribo and B. Pesquet-Popescu, "Depth-aided image inpainting for novel view synthesis," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Oct. 2010, pp. 167–170.

[16] C. Yao, Y. Zhao, J. Xiao, H. Bai, and C. Lin, "Depth map driven hole filling algorithm exploiting temporal correlation information," *IEEE Trans. Broadcast.*, vol. 60, no. 2, pp. 394–404, Jun. 2014.

[17] M. Paul, W. Lin, C.-T. Lau, and B. S. Lee, "A long-term reference frame for hierarchical B-picture-based video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, pp. 1729–1742, Oct. 2014.

[18] C. Zhu and S. Li, "Depth image based view synthesis: New insights and perspectives on hole generation and filling," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 82–93, Mar. 2016.

[19] P. Pandit, A. Vetro, and Y. Chen, *Joint Multiview Video Model (JMVM) 7 Reference Software*, document N9579, MPEG of ISO/IEC JTC1/SC29/WG11, Antalya, Turkey, Jan. 2008.

[20] M. Talebpourazad, "3D-TV content generation and multi-view video coding," Ph.D. dissertation, Dept. Electr. Comput. Eng., Univ. British Columbia, Vancouver, BC, Canada, 2010.

[21] S. Yea and A. Vetro, "View synthesis prediction for multiview video coding," *Signal Process., Image Commun.*, vol. 24, nos. 1–2, pp. 89–100, Jan. 2009.

[22] S. Ma, S. Wang, and W. Gao, "Low complexity adaptive view synthesis optimization in HEVC based 3D video coding," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 266–271, Jan. 2014.

[23] B. T. Oh and K.-J. Oh, "View synthesis distortion estimation for AVC- and HEVC-compatible 3-D video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 1006–1015, Jun. 2014.

[24] A. I. Purica, E. G. Mora, B. Pesquet-Popescu, M. Cagnazzo, and B. Ionescu, "Multiview plus depth video coding with temporal prediction view synthesis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 2, pp. 360–374, Feb. 2016, doi: 10.1109/TCSVT.2015.2389511.

[25] H. Yuan, Y. Chang, J. Huo, F. Yang, and Z. Lu, "Model-based joint bit allocation between texture videos and depth maps for 3-D video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 4, pp. 485–497, Apr. 2011.

[26] P. Gao and W. Xiang, "Rate-distortion optimized mode switching for error-resilient multi-view video plus depth based 3-D video coding," *IEEE Trans. Multimedia*, vol. 16, no. 7, pp. 1797–1808, Nov. 2014.

[27] M. Haque, M. Murshed, and M. Paul, "Improved Gaussian mixtures for robust object detection by adaptive multi-background generation," in *Proc. IEEE Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.

[28] D.-S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005.

[29] M. Paul, "Efficient multi-view video coding using 3D motion estimation and virtual frame," *Neurocomputing*, vol. 175, pp. 544–554, Jan. 2016.

[30] X. Xu and Y. He, "Fast disparity motion estimation in MVC based on range prediction," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 2000–2003.

[31] J. Seo and K. Sohn, "Early disparity estimation skipping for multi-view video coding," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, p. 32, Dec. 2012.

[32] X. Guo, Y. Lu, F. Wu, and W. Gao, "Inter-view direct mode for multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 12, pp. 1527–1532, Dec. 2006.

[33] J. Konieczny and M. Domański, "Depth-based inter-view prediction of motion vectors for improved multiview video coding," in *Proc. 3DTV-Conf., True Vis.-Capture, Transmiss. Display 3D Video*, Jun. 2010, pp. 1–4.

[34] H.-S. Koo, Y.-J. Jeon, and B.-M. Jeon, *Motion Skip Mode for MVC*, document ITU-T and ISO/IEC JTC1, JVT-U091, 2006.

[35] F. Shao, G. Jiang, M. Yu, K. Chen, and Y.-S. Ho, "Asymmetric coding of multi-view video plus depth based 3-D video for view rendering," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 158–167, Feb. 2012.

[36] *View Synthesis Reference Software 3.5*, document ISO/IEC JTC1/SC29/847 WG11 (MPEG), 2013.

[37] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD Curves*, document ITU-T SC16/Q6, VCEG-M33, Austin, 2001.

[38] P. K. Podder, M. Paul1, and M. Murshed, "A novel motion classification based intermode selection strategy for HEVC performance improvement," *Neurocomputing*, vol. 173, pp. 1211–1220, Jan. 2016.

D. M. Motiur Rahaman (S'16) received the B.Sc. degree in electrical and electronic engineering from the Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh, in 2011. He is currently pursuing the Ph.D. degree with Charles Sturt University, Australia, as Research Scholar. He joined as a Lecturer with the Department of Electrical, Electronics and Telecommunication Engineering, Dhaka International University, Dhaka, Bangladesh, since 2011. His research interest resides in the field of computer vision, image processing video coding artificial intelligence, and affective computing. He has authored several refereed papers/articles in this field. He is a student member of the Program Committee of Pacific-Rim Symposium on Image and Video Technology 2017.

Manoranjan Paul (M'03–SM'13) received the Ph.D. degree from Monash University in 2005. He was a Research Fellow with the University of New South Wales, Monash University, and Nanyang Technological University. He is currently an Associate Professor with the School of Computing and Mathematics, Charles Sturt University. He authored or co-authored over 150 refereed papers in international journals and conferences, and journal articles in the IEEE TRANSACTIONS. His major research interests are in the fields of image/video coding, EEG signal analysis, and computer vision. He is a Senior Member of the Australian Computer Society (ACS). He was a recipient of $15 million competitive grant including the Australian Research Council Discovery Project and the Australian Government's CRC Project grant. He served as a Guest Editor of the *Journal of Multimedia* and the *Journal of Computers* for five special issues and the Publicity Chair of DICTA 2016 and the Program Chair of PSIVT 2017. He has been an Associate Editor of the *EURASIP Journal on Advances in Signal Processing* since 2013. He was the ICT Researcher of the Year 2017 by ACS. He was a keynote speaker in the IEEE DICTA-17, WoWMoM-14 Workshop, DICTA-13, and ICCIT-10.