Anantrasirichai, N., Canagarajah, C. N., Redmill, D. W., Akbari, A. S., & Bull, D. (2011). Colour volumetric compression for realistic view synthesis applications. Multimedia Tools and Applications, 53(1), 25-51. 10.1007/s11042-010-0484-4

Colour Volumetric Compression for Realistic View Synthesis Applications

N. Anantrasirichai, C. Nishan Canagarajah, David W. Redmill, Akbar Sheikh Akbari, David R. Bull

University of Bristol, Woodland Rd., Bristol, UK, BS8 1UB
Contact Email: n.anantrasirichai@bristol.ac.uk

Abstract— **Colour volumetric data, which is constructed from a set of multi-view images, is capable of providing realistic immersive experience. However it is not widely applicable due to its manifold increase in bandwidth. This paper presents a novel framework to achieve scalable volumetric compression. Based on wavelet transformation, data rearrangement algorithm is proposed to compact volumetric data leading to high efficiency of transformation. The colour data is rearranged using the characteristics of human visual system. A pre-processing scheme for adaptive resolution is also proposed in this paper. The low resolution overcomes the limitation of the data transmission at low bitrates, whilst the fine resolution improves the quality of the synthesised images. Results show significant improvement of the compression performance over the traditional 3D coding. Finally, effect of using residual coding is investigated in order to show a trade off between the compression and view synthesis performance.**

Index Terms—Multi-view Image, Volumetric representation, View synthesis, Residual coding

1. INTRODUCTION

Three-dimensional (3D) visual environments have become one of the emerging technologies in communication and computer vision. Multiple cameras are used to capture scene information from various points of view. Its large amount of data has limited many of its commercial applications, due to its manifold increase in bandwidth over existing monoscopic compression technologies. *Volumetric representation* of the 3D data is a good solution to reduce the required bandwidth, since it can effectively compress the data at high compression ratios whilst preserving an accurate model of the object. However, efficient compression algorithms are vital to reduce the size of the volumetric data without sacrificing the quality of the images. Compression of the constructed volume can be efficiently achieved by exploiting 3D and/or geometry coding schemes. The 3D wavelet-based algorithms for volumetric compression have been developed with high efficiency for both lossless and lossy coding in many researches [1][2][3]. However, most of these algorithms have been proposed for medical applications of which details are 2D cross sections of the volumes, rather than view synthesis.

Direct coding of 3D voxel models has been carried out by using the idea of video coding in which each third component is operated as a frame of the sequence [4]. For view synthesis applications, this scheme has unsuccessfully been applied at low bitrates. Due to lossy compression, some model information may be lost which cause artefacts and lack of details in the synthesised images. To overcome these weaknesses, we propose to encode the model and colours of the volume separately. Thus, the model is losslessly coded so that the correct geometry is obtained even at very low bitrate, whilst the colour information is possibly truncated to achieve the target bitrate. This approach increases the degrees of freedom for designing the algorithm in order to enhance compression efficiency. Other compression algorithms have been proposed in [5], [6] and [7], i.e. geometry compression which is one of the 3D data compressions provides the scene description to render images by coding vertex positions. In [8], it was shown that the vertex coding can be improved by exploiting the correlation among the adjacent surface voxels. Although these mesh-based coding schemes, which exploit 3D geometric information, greatly outperform the conventional 2D predicting compression, they are highly complicated and time-consuming [9]. Moreover, the mesh-based coding is difficult to apply to the complex shapes and surfaces which generally present in natural scenes.

Apart from medical applications, the inside of the object is generally occluded and is not essential for view synthesis. Existing methods introduced for compressing volumetric data contain this useless information in the coding process causing inefficiency in data transmission and storage. To achieve the highest compression performance, two data rearrangement schemes are introduced in this paper; one uses the characteristics of human eye sensitivity to diminish the size of chrominance information and the other prepares the dense data for 3D wavelet transformation. Moreover, we propose the adaptive resolution approach for the volume representation which provides scalabilities and increases view synthesis performance. By using this approach, the receiver can use only the significant part of the bitstream instead of the entire bitstream. Consequently, images at full resolution with acceptable quality can be constructed using the half-resolution volume. If the receiver can support bigger data size or a bigger buffer, the double-resolution volume can be used for constructing images at higher quality.

In this paper, a novel volumetric compression algorithm which is suitable for realistic view synthesis is presented. It first constructs the volume with adaptive resolution and prepares the colour data to improve the coding performance. The colour data is then transformed using a wavelet lifting scheme. The transformed data is finally encoded using the JPEG2000 Part-2 standard [10]. This compression standard supports multiple compositing layers which is extended from the JPEG2000 Part-1 [11], leading to the better performance for volumetric data [12]. This coding method was selected for two reasons: i) the wavelet transform provides great performance with enhanced features, i.e. scalabilities; ii) the codec following the standard gains the compatibility and interoperability.

The rest of paper is organized as follows: Section 2 explains the volume reconstruction with registration method. Section 3 discusses the proposed schemes for the volumetric representation with adaptive resolution. The proposed codec is described in Section 4. Experimental results and discussions are presented in Section 5. The bit allocation to optimise the desirable compression and view synthesis performance is also investigated in this paper. The residual coding is added to the proposed codec in order to enhance the quality of the reconstructed images. The effect of

residual coding on view synthesis performance is then investigated in Section 6. The conclusions and future works are presented in Section 7.

2. VOLUMETRIC REPRESENTATION

To construct a volume, the volumetric representation algorithm proposed in [13] is exploited. A sparse non-Lambertian multi-view system where all the cameras can be different, and may converge to different points in the scene is considered. The proposed volume construction includes three steps: Volume initialization, Volume refinement and Colour selection, which will be discussed in the following sections.

2.1 Volume Initialization

We construct the volume by registering the estimated depth maps. Not only is this registration simple and fast, but it also provides a degree of freedom to select an appropriate depth estimation approach. Each view along its own geometry is searched for the depth map. Therefore it is possible to compensate non-Lambertian reflection by exploiting the individual minimum cost instead of defining a global threshold to obtain the depth value. It is well known that the depth registration methods give a good representation for the natural scenery [14]. Depth maps are registered into a volume. Each registered voxel has a value equal to the number of views that are mapped to this voxel; therefore, the unmapped voxels have a value of 0. The voxel value $\delta_{\widetilde{ijk}}$ can be expressed as follows;

$$\delta_{\widetilde{ijk}} = \begin{cases} n, & \textit{if registering depth}, \\ 0, & \textit{otherwise} \end{cases} \qquad (1)$$

where the number of views that are mapped to a voxel $\tilde{\mathbf{x}}$ is n for $n \in \{1, 2, ..., N\}$, with total number of N reference views, and $\tilde{i}, \tilde{j}, \tilde{k}$ show the locations of the voxel in a 3D volume.

After calculating the volume, some isolated voxels which do not have any neighbours are detected. The volume is then projected back to the reference view geometry to build up new depth maps. The resulting depth maps include the depth information from other views that some errors have been removed in the previous step. If there is more than one depth value for a pixel, the front cluster of depths is considered. There is ideally one depth value projected from the front surface, but spurious noise makes it dispersed. The proper depth value $\tilde{d}$ of a particular pixel is

$$\tilde{d} = \begin{cases} d_n, & \textit{if only one depth has } \delta_n = \max(\delta), \\ \left(\sum_{k=1}^{K} \delta_k d_k\right) / \left(\sum_{k=1}^{K} \delta_k\right), & \textit{otherwise} \end{cases} \qquad (2)$$

where K is the number of depth values in the front cluster. The depth value projected from the voxel that contains

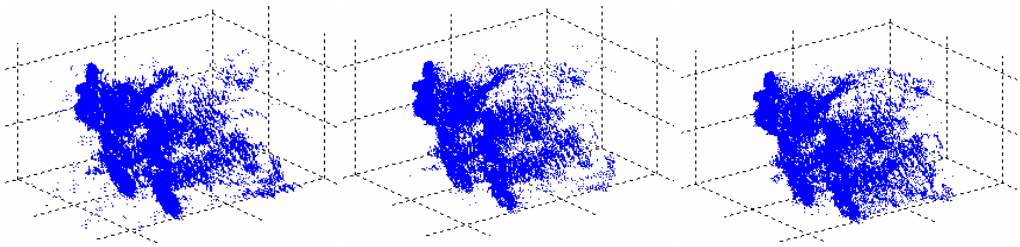
the maximum δ is selected. If the maximum δ appears in two or more depth values, the fit depth value is calculated from the average of all depth values of the front cluster with weighted coefficient δ. Subsequently, the volume is reconstructed again. In this step, some voxels are interpolated in order to connect the surface of the volume among the near voxels.

2.2 Volume Refinement

In this section, a novel algorithm to refine the shape of the volume is proposed. The constructed volume should accurately represent the real scene and object, as any errors in the constructed volume could adversely affect the virtual view synthesis. However, sharp corners always reduce the achievable coding gain; therefore, the sharp corners of the surface are smoothed by considering the neighbouring voxels.

We reconstruct all views from the volume with colours and compare them with the original reference images. The colours are temporarily defined by taking the median of colour values from all the visible views. The iterative process runs from the first reference view, probably the furthest left view, to the last reference view, probably the furthest right view. For each view, the mean square errors (MSE) between the reconstructed view and the original view are calculated. If the error is more than a threshold, the old depth is removed and it also causes the associated voxel to be removed. Then the new depth, which is derived from the neighbouring pixels (a smooth surface assumption), is investigated. If this new depth value meets colour constraints for all visible views, the new associated voxel is generated. It is noticeable that inaccurate camera calibration and unequal resolutions of each of the reference views possibly displace a mapped pixel from its exact location. A window-based computation algorithm (e.g. 3x3 pixels) gives better results for error calculation than exploiting only one pixel.

If some voxels are removed or newly generated, we iteratively reconstruct the images. As any changes in the position of a voxel produces a new depth, which might change an invisible point to a visible point and vice versa. The initial volume illustrated in Fig. 1 (a) is generated by registering the estimated depth maps. It can be seen that there are a lot of scattering voxels in the initial volume, but it is significantly improved by removing the scattering voxels using the proposed scheme as shown in Fig. 1 (b). The volume is iteratively refined until no more new voxels are generated and no more old voxels are removed (26 iterations for five-view *Leo* sequence). The final volume is shown in Fig. 1 (c).



(a) (b) (c)

Fig. 1 The voxel clouds after (a) depth maps were registered; (b) separating voxels were removed; (c) iterative volume refinement.

2.3 Colour Selection

Finally, colours are defined for each voxel. To ensure a smooth *look around* capability, a good colour selection method should compensate the noise from several sources, e.g. camera calibration and imaging, imperfect light balance and volume quantization. The weighted average of the colours from one or two of the nearest visible views are used for defining voxel colour $\tilde{C}$, leading to smooth colours and brightness. The colours of voxels are determined using Equation 3 with a *condition* that the voxel is visible at least in two views.

$$\tilde{C} = \begin{cases} (\alpha_1 C_2 + \alpha_2 C_1)/(\alpha_1 + \alpha_2) & \textit{if the condition is true}, \\ C_1 & \textit{otherwise} \end{cases} \tag{3}$$

where C_1 and C_2 represent the intensity component (Y) or the colour components (U,V) of view 1 and 2, respectively. α_1 and α_2 are the angles between the current voxel and the point in the reference view 1 and view 2 respectively.

Using this method for defining the voxel colour may cause object to be blurred, if C_1 and C_2 are too different. Therefore, the defined colour is set into a *blur* group, if the difference of C_1 and C_2 is more than a particular threshold; otherwise, are set into a *clear* group. To preserve the sharpness of the volume, the colour $\tilde{C}_{old}$ of the voxel in a *blur* group is replaced by colour of neighbouring voxel in the clear group that has the closest value to $\tilde{C}_{old}$.

3. PRE-PROCESSING FOR ADAPTIVE RESOLUTION

The adaptive resolution approach for the volumetric representation which provides scalabilities and increases view synthesis performance is described in this section. Through this approach, the receiver can use only the significant part instead of the entire data stream. Consequently, full-resolution images can be constructed from the half-resolution volume with acceptable quality. If a receiver can support bigger data size or a bigger buffer, the double-resolution volume is employed to construct images with higher quality. That is, the final results are the double-resolution volume which is capable of providing good image quality when the volume is used as full or half resolution. In order to give priority to the original image size, the full-resolution volume is first constructed using the algorithm proposed in section 2. The existing voxels which are parts of the object contain intensity *Y* and colours *U* and *V*, whilst the voxels which are outside the object or invisible are transparent. The proper voxels are then inserted in a place to complete the volume, when it is used at smaller or larger scale.

3.1 Pre-processing for Half Resolution

For lower resolution, subsampling causes significant information loss; therefore, a pre-processing step is introduced

as illustrated in Fig. 2. By subsampling, the voxels at location $(2\hat{i}-1, 2\hat{j}-1, 2\hat{k}-1)$, where $1 \le 2\hat{i}-1 \le h$, $1 \le 2\hat{j}-1 \le w$, $1 \le 2\hat{k}-1 \le z$, and $\hat{i}, \hat{j}, \hat{k} \in \{1,2,3,...\}$ are included in the half-resolution volume. Subsequently, every position containing the transparent voxels is checked. If one of its eight adjacent voxels of the full-resolution volume is not transparent, the transparent voxel at location $(2\hat{i}-1, 2\hat{j}-1, 2\hat{k}-1)$ is replaced by the median of the existing voxels at these eight adjacent positions. The new voxels which are located outside the volume are then removed to ensure that the new inserted voxels do not occlude or affect any point of the original volume at full resolution. The pre-processed data contains more voxels, hence the holes in the synthesized views are disappeared and the quality of the resulting image is improved.

When the decoder receives the first part of the bitstream, the half-resolution model is reconstructed, and then the colour volume is reconstructed. The reconstructed images or the virtual views with the half size of the original images can be generated by directly employing this half-resolution model and the colour volume. However, to create a full-resolution image, the 3D interpolation process is required. The full-resolution reconstructed image from the half-resolution volume is shown in Fig. 3 (b). Comparing to the post-processing interpolation from 2D images shown in Fig. 3 (c) and Fig. 3 (d), the reconstructed image from the 3D interpolation of the volumetric representation presents superior quality. It achieves sharpness at object's edges and smoothness at homogenous areas. It also achieves the highest PSNR, 32.08 dB, while other approaches gain 22.75 dB for Fig. 3 (a), 27.44 dB for Fig. 3 (c) and 26.03 dB for Fig. 3 (d).

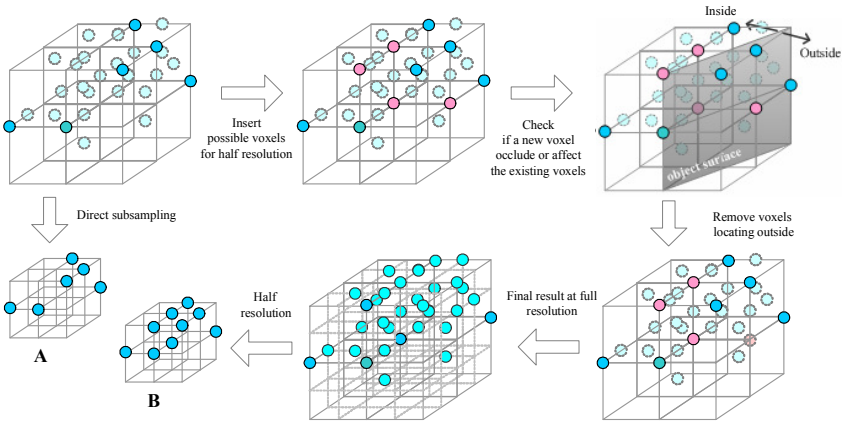
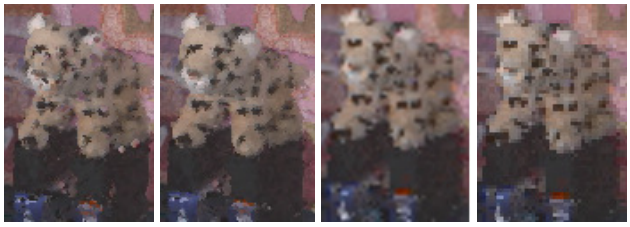


Fig. 2 Half resolution process



(a) (b) (c) (d)

Fig. 3 The reconstructed image at full resolution by interpolating from the half-resolution volume: (a) without preparation process/direct subsampling (A); (b) with preparation process (B), and interpolating from the half-resolution 2D image with (c) linear interpolation and (d) nearest interpolation.

Comparing to the wavelet transformation, the subsampling from the prepared volume produces better shape contours at small scale. This is because filtering the binary data of the model possibly generates the floating-point data which is hard to define whether it is the existing or transparent voxel. Moreover, high different values between the transparent voxels and the existing voxels generate high frequencies. Therefore, details are significantly lost if only low-pass signal of the transformed volume is used. As a result the synthesised views at low bitrate possibly have poor perceptual quality.

3.2 Pre-processing for Double Resolution

To increase the resolution, the voxel at location (i, j, k) of the full-resolution volume is mapped to the location ($2i$-1, $2j$-1,$2k$-1) of the double-resolution volume. The other voxels are inspected to determine whether they are part of the volume or transparent by operating the same construction approach of the full-resolution volume. That is, the depth maps of the references are first registered into the double-resolution volume. The constructed volume is then iteratively refined by utilizing the brightness information of the reference views. Finally, the colour of each voxel is properly selected in order to render a smooth and realistic image. Fig. 4 illustrates the proposed double resolution process.

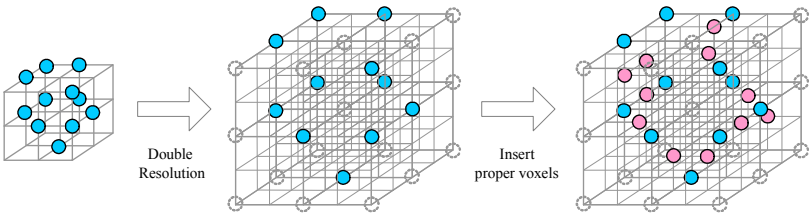


Fig. 4 Double resolution process (● = existing voxel, ○ = non-existing voxel, ● = new added voxel).

The finer-resolution volume improves the view synthesis performance, because it provides more details along with more accurate voxel positions. As the imaging systems operate in discrete domain the computed positions are adjusted to their nearest integers. Some non-corresponding pixels from the different reference views, which have

different spatial resolutions, are possibly mapped to the same voxel. Consequently, the synthesized views might obtain the absolutely wrong values of brightness and colours leading to a poor subjective quality if the resolution of the representing volume is not fine enough. In contrast, the double-resolution volume allows the existing voxel to be located at the middle point between two voxels of the full-resolution volume leading to precise shape and contour of the object.

4. PROPOSED CODING SCHEME

After generating the 3D volume from a set of multi-view images, the model and texture are encoded to reduce the data size with the highest perceptive quality. The compressed data is more suitable for storage and transmission in a band limited transmission channel. In addition, the coding scheme should provide scalabilities in order to support heterogeneous networks and instruments.

There are two major parts of the volumetric data: i) the texture T which comprises the luminance component Y and the chrominance components U and V, and ii) the model m which identifies the position of each voxel in a 3D volume. The model m is the binary volume of which the value '1' identifies an existing voxel and the value '0' identifies a transparent voxel. The general diagram of the proposed volumetric coding algorithm is illustrated in Fig. 5. It can be seen from the diagram that the luminance Y is utilised to identify the model structure. Then, the texture T of each resolution level is processed independently. The luminance Y is input to the depth-plane reduction process, whilst the chrominance U,V are subsampled before entering the depth-plane reduction block.

The model is losslessly and independently coded from the texture, since the depth or the position of each voxel is vital for image warping and view synthesis. As each voxel represented by 24 bits, it is impractical to code the texture information losslessly. Therefore, the texture information is lossy coded. If the lossy 3D coding directly applies to the volumetric data, the loss of any volume information affects the reconstruction of the model. In contrast, separately coding the model produces a perceptible benefit at low bitrates, because the decoder receives the true model which is then employed for rendering images with the correct shapes.

After compressing the model, the luminance component Y and the chrominance components U and V are reorganised into more appropriate forms so that the compression method can attain the highest efficiency. The lossy coding is subsequently applied to achieve the target bitrates. Although a loss of colour information is shown, the subjective quality of the reconstructed image when independently coding the model is better than that of the traditional 3D coding scheme as illustrated in Fig. 6. The ambiguity of the existing decoded voxels is absent as clearly shown in the depth map in Fig. 7.

Once the model and the texture are coded at a particular bitrate, the data stream is organized as follows. The half resolution volume data is firstly sent to support the very low bitrate applications. It is then followed by the additional data in the full resolution excluded from the previous transmission. Finally, the remaining data used for composing the double-resolution volume is included in the data stream. Noticeably, this scheme provides scalability. However,

the interpolation may be needed to enlarge the rendered images when a half-scale volume is used. The camera parameters are separately coded using Huffman coding, and stored or transmitted as the side information of the data stream. The background of the same geometry of the volume is independently coded from the volumetric data. The benefit of this algorithm is that it allows bits to be straightforwardly allocated when coding the region of interest. The background can also be coded with less number of bits and also a different background can be used depending on the users/applications.

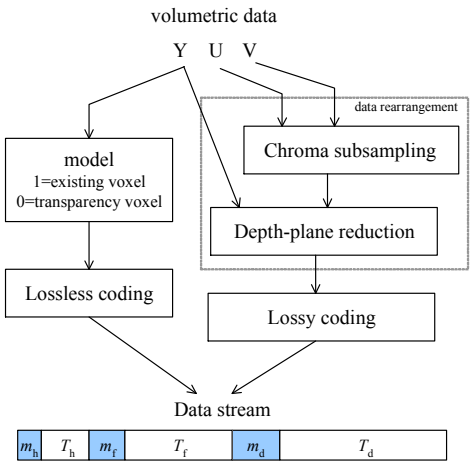
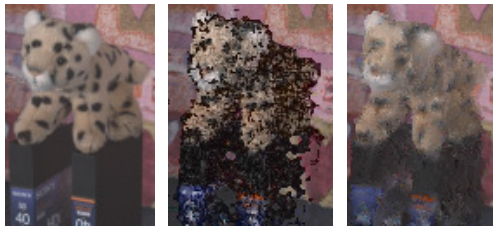
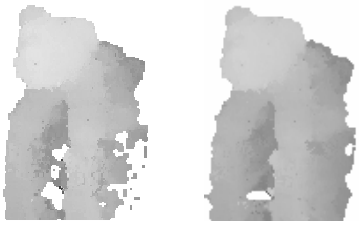


Fig. 5 Coding diagram of the volumetric compression (m_x=coded model, T_x=coded texture, $x \in \{h, f, d\}$, h=half resolution, f=added to full resolution, d=added to double resolution)



(a) (b) (c)

Fig. 6 (a) The original image and the reconstructed image at 200kbits (b) without model coding. (c) with model coding.



(a) (b)

Fig. 7 The depth maps of images in Fig. 6 corresponding to the reconstructed image (a) without model coding. (b) with model coding.

4.1. Data Rearrangement

In this paper, two data rearrangement approaches are proposed, called chroma subsampling and depth-plane reduction. The first approach reduces the resolution of chrominance and the other approach prepares the dense data which is suitable for 3D wavelet transform.

4.1.1 Chroma Subsampling

Due to the fact that the human's eyes are less sensitive to colour than brightness, the chrominance of an image can be defines with lower resolution than the luminance. However, for the volume, the intensity and colour are defined only for the existing voxels. Direct subsampling of the volume causes significant information loss. To preserve good colour information at lower resolution, one colour value for every eight adjacent corners is defined. It is taken as a transparent voxel if there is no existing voxel in any of its eight corners. If one or more voxels are presented in any of the eight adjacent corners, the median value of the existing voxels are used as the colour value as shown in Fig. 8. The median method is used since it gives better sharpness than the average method. It is also less influenced by outlier values, so the high different values are ignored.

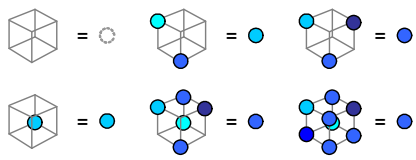


Fig. 8 The examples of chroma subsampling in various cases (○ = the non-existing voxel, ● = medium value of all existing voxels).

In the decoder side, the missing values are interpolated from the neighbouring voxels or repeated from the preceding sample. Since viewers' sense of colour is barely affected, the repeating approach, is employed in our simulation.

4.1.2 Depth-plane reduction

The volume reconstructed from multiple camera views is formed with the object's surface in which the inside is generally occluded and is not essential for the view synthesis. Compared to a solid volume where the inside voxels are filled in order to reduce the high frequencies after transformation, the number of voxels along the surface is much smaller than those of the whole object. Therefore, instead of coding a solid volume with the original shape, the volume is deformed to reduce the number of depth planes by shifting the voxels forward to fill the empty depth plane in each width. The deformation approach is applicable since the model is coded independently. Finally, the

non-existing voxels are filled with the value of 128. This constant value is selected because it is the centre value which will be shifted to zero in the pre-processing step of the JPEG2000 codec. Consequently, the number of depth planes is reduced by using this approach, and the holes also disappear leading to huge reduction of energy dispatched to high frequency subbands after applying wavelet transform. The proposed algorithm is simply explained by pictures illustrated in Fig. 9.

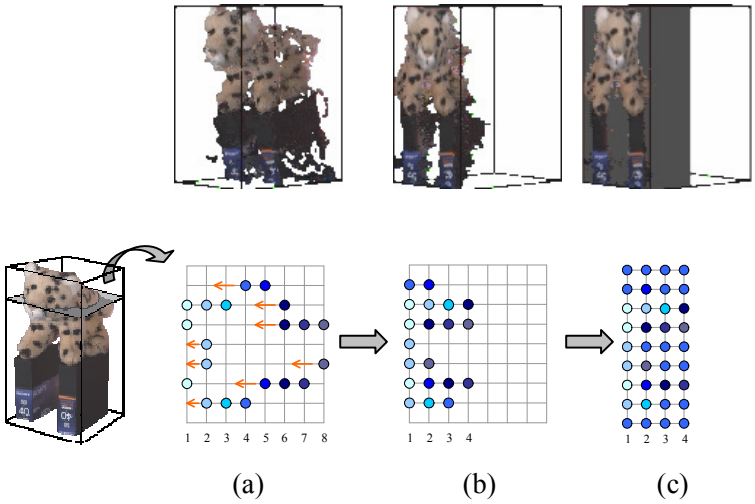


Fig. 9 The example of a cross section of a volume showing data rearrangement idea. (a) original volume with 8 components (b) rearranged volume to 4 components (c) fill in all non-existing voxels with the medium value.

In the decoder side, the model is used for identifying the position of each voxel. The original position of the shrunk luminance and chrominance data are retrieved, by reversing the chroma subsampling and depth-plane reduction processes.

4.2 Wavelet-based Encoder

A 3D volume is an image where one sample is composed of multiple components. This characteristic meets the definition of the JPEG2000 Part-2 standard. Thus the multi-component transformation can be employed. The wavelet lifting scheme is applied to each pixel in the image, as well as that in the third direction.

The lifting based wavelet transform is used to realize a fast wavelet transform. Since two lines are processed at a time, this architecture allows minimum memory requirement and fast calculation. This method decomposes wavelet transform into a set of stages. The operation starts with a split step, which divides the data set into two groups. The next step is prediction where one group of data is used to predict other group of the data. High-pass residual signal, H_i, is then generated by subtracting the predicted elements from the original elements. Therefore, the high-pass signal contains small amount of energy, which increases the achievable compression efficiency. Finally an update step combines residual data from the previous stage to reduce the effect of aliasing in low-pass signal, L_i.

In the proposed scheme, the new compact volumes are transformed to wavelet coefficients. Along the component axis, the Haar and 5/3 wavelet lifting schemes are applied to remove correlations among consecutive components of the luminance Y and chrominance U and V, respectively. Basically, the longer filters achieve the higher compression efficiency but, in the case that the correlations among the successive components are not high enough and/or the total number of components is too small, the longer filters generate more energy at high frequencies and will cause the ghosting artefacts. A short filter, such as Haar, is therefore selected for the luminance Y rather than the 5/3 wavelet which is used for transforming the chrominance U and V. After applying the multi-component transformation with particular number of levels, each frame is compressed using the JPEG2000 Part-1 standard. Fig. 10 illustrates three examples of the 3D wavelet transform. The volume is decomposed by three levels and four different types of frames are generated, called low-low-low-pass frames (LLL), low-low-high-pass frames (LLH), low-high-pass frames (LH) and high-pass frames (H). Subsequently, each frame is spatially decomposed via 2D wavelet lifting scheme and coded with EBCOT coding. These examples display various levels of the wavelet transform for each subband frame. However, the experimental results show a small difference in their coding performance.

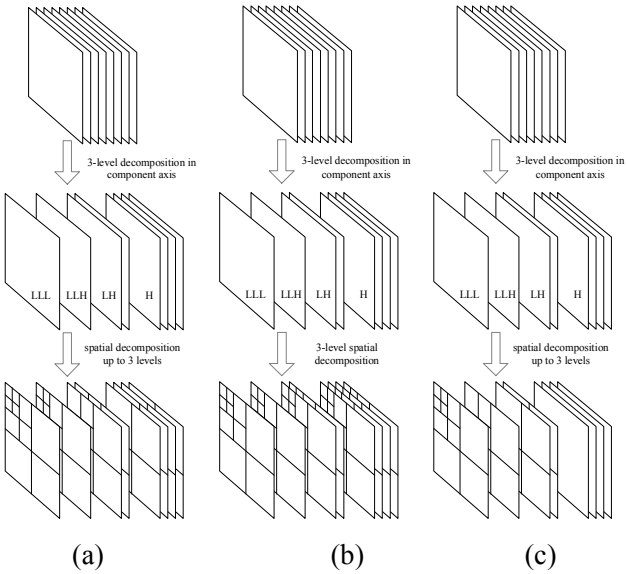


Fig. 10 Three examples of 3D wavelet transformation; (a) all decomposed depth planes are further spatially transformed with 3 levels, (b) the high-pass components are further spatially transformed only one level, whilst the low-pass components are transformed up to 3 levels, (c) the high-pass components are not transformed, but the low-pass components are transformed up to 3 levels.

The JPEG2000 JP3D standard (part 10) can also be used for compressing 3D data. It is supposed to provide better performance but this part of standard is still at the Working Draft stage [15]. The use of the extensions (part 2) is sufficient to show the improvement of the proposed schemes, i.e. preparation for adaptive resolution and data rearrangement techniques.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

The proposed scheme is tested with the volume of *Leo*[1] sequence that is constructed by exploiting the proposed volume reconstruction as described in section 2. This multiview sequence is composed of five views with different camera parameters that cause different spatial resolutions. The background of this sequence is separately coded but shares the bitrate. This multiview images composes of five views with different camera parameters that cause different resolutions. The following subsection investigated the proposed schemes with fix and adaptive resolutions.

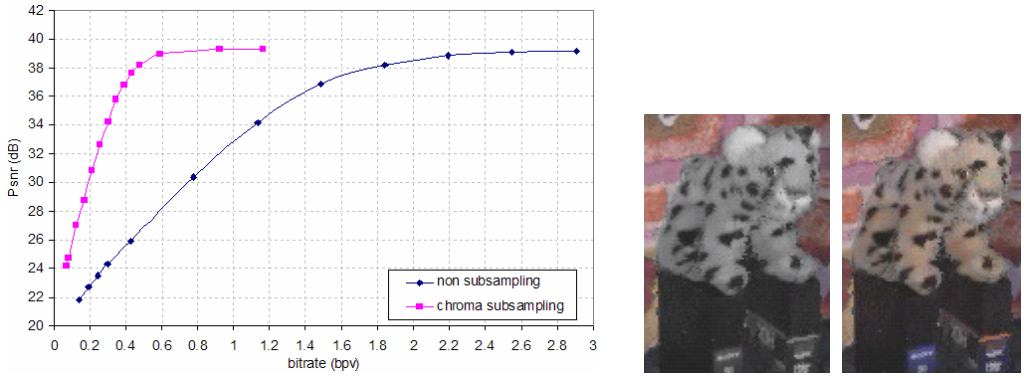
5.1 Compression Performance

5.1.1 Fix Resolution

In this section, one-pixel precision of the middle view was used for identifying the resolution of the constructed volume for *Leo* sequence which is 128x80x86 (height x width x depth). The quality of five reconstructed views was measured using their average PSNR. The background of each view was regenerated using the estimation depth map. The estimated depth maps was generated using dynamic programming algorithm proposed in [16]. For this section, the background was not encoded, so bitrates shown in the graphs comprise only the information required to reconstruct the volume, i.e. binary model, brightness and colour of texture surface and camera/geometry parameters. Note that the coding performance depends on the accuracy of the available geometry model [9].

The proposed chroma subsampling was first tested and compared to the non-subsampling approach. Only the performance of the chrominance components U, V was investigated, so the bitrates shown in the graph does not include bits to represent the luminance component Y. The experimental results shown in Fig. 11 reveal that the chroma subsampling improves the quality of the reconstructed colour components. The quality of the chroma-subsampling approach reaches the best achievable quality, 39.30 dB, at 1.18 bpv (bits per voxel), whilst that of the non-subsampling approach attains the quality of 39.30 dB at 2.90 bpv. The subjective results in Fig. 11 (b-1) prove that the chroma information can be completely lost at low bitrates if the chroma subsampling is not applied as the object is grey, whereas the reconstructed object with chroma subsampling (Fig. 11 (b-2)) still contains colours when it is coded at the same bitrate.

[1] The multiview *Leo* sequence was captured at University of Bristol



(a) (b-1) (b-2)

Fig. 11 Comparison results between subsampling and non-subsampling of colour components; (a) the objective results and (b) the subjective results coded at 0.2 bpv (1) without chroma subsampling and (2) with chroma subsampling.

The proposed depth-plane reduction is considered in this stage. As a result, the number of depth planes of the *Leo* volume was reduced from 86 to 25. Fig. 12 shows the average PSNR of the Y component of the five reconstructed views. It displays three different experiments: i), the traditional approach which directly used the JPEG2000 Part-2 coding; ii), the depth-plane reduction approach without filling the non-existing voxel space; and iii) the depth-plane reduction approach with filling the empty space. The plot shows that the quality of the reconstructed images from the data rearrangement suddenly drops at around 0.14 bpv, since some amount of bits is allocated for coding the model/shape of the volume. However, when the bitrate slightly increases, the image quality significantly improves and reaches the best achievable quality of 33.08 dB at the 0.63 bpv. This bitrate is significantly less than the bitrate of the traditional approach which attains the quality of 33.08 dB at 1.25 bpv. This indicates that the smaller number of depth planes requires smaller number of bits, although the number of the existing voxels are identical. Noticeably, the number of depth planes is reduced over three times, while the number of bits used for coding the modified volume decreases only half of the bits for coding the original volume. This is because the energy per depth plane of the modified volume is higher than that of the original volume. The experimental results also show that filling the non-existing voxels with the value of 128 can improve the quality of the reconstructed images of up to 1.4 dB, since energy in the high frequency component is reduced and most energy of the image moves to the baseband.

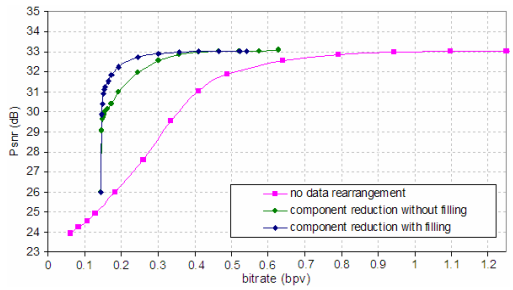


Fig. 12 The performances of the proposed depth-plane reduction scheme and the filling method by the median values of the existing voxels.

5.1.2 Adaptive Resolution

Experimental results in section 5.1.1 show that the proposed scheme has some limitations at very low bitrates as it requires a minimum number of bits to code the model. Therefore, the scalability of the model coding is needed so that some part of the data stream can be used for reconstructing the model with an acceptable quality rather than using the whole data stream.

In this section, three options of volume resolutions are considered: half, full and double resolutions. The lower resolution with 2-pel precision eliminates the limitation at low bitrate. The coding at bitrates smaller than the minimum bits needed for lossless coding of the whole model is possible. However, the interpolation is exploited to enlarge the volume as explained in section 3.1. The higher resolution with half-pel precision is involved in order to enhance the image quality and view synthesis performance.

As illustrated in Fig. 13, the maximum quality of the half-resolution volume is approximately 1.1 dB less than that of the full-resolution volume, whilst the double-resolution volume improves the compression performance up to 0.3 dB. This is because the more details and higher accuracy are obtained from the finer resolution. The details of the volume cannot cover every pixel in the projected images if the resolution of the volume is coarser than that of the reconstructed images. In this case, the holes appeared in the images are filled by interpolation using the neighbouring voxels instead of being directly rendered from the finer-resolution volume. Fig. 14 shows the subjective results of the double-resolution reconstructed views rendered from (a) the full-resolution volume, and (b)-(c) the double-resolution volume. The double-resolution volume illustrated in Fig. 14 (b) comes from 3D interpolation of the full-resolution volume, whilst the double-resolution volume shown in Fig. 14 (c) is constructed by registering the depth maps and using the scheme proposed in section 3.2. The latter result shows significant quality. The errors in the reconstructed view of the first two cases are more vivid due to interpolation.

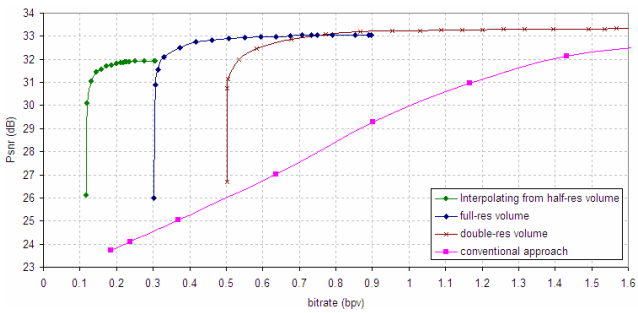


Fig. 13 The performance of thee-resolution volume compared to the conventional approach which does not exploit any enhanced technique.

(a) (b) (c)

Fig. 14 The double-resolution reconstructed images (a) projected from the full-resolution volume with linear interpolation. (b) projected from the double-resolution volume which is 3D interpolated from the full-resolution volume. (c) projected from the proposed double-resolution volume.

The *Head* layered image comprising 7 layers was also used to investigate the performance of the proposed scheme[2]. Basically, the layered image comprises many 2D image planes; therefore, the compression performance of the proposed technique was compared to the individual layered coding algorithm instead of the conventional 3D approach. That is, the JPEG2000 Part-1 was applied to code each layer separately. The bitrate in the demonstration graph is the total bits used for compression rather than the bpv.

The proposed scheme improves the quality of the reconstructed images of up to 5 dB as shown in Fig. 15. The improvement of the quality of the reconstructed layered images is less than the improvement of the quality of the volume images, since the correlation among the successive layers is less than the correlation in the volume images. The reconstructed images encoded at 35 kbits/view are illustrated in Fig. 16. The results of the individual coding approach (Fig. 16 (a-1)) are affected from the lack of model information, hence the rendered results show the dark colour around the layer boundaries. The PSNR of the half-resolution volume is less than that of the full-resolution volume, but the subjective results of the half-resolution volume show higher texture quality. This is clearly shown in the magnified version of the reconstructed *Head* images, which is illustrated in Fig. 16 (b). The PSNR of the half-resolution volume is less than the finer-resolution volume because of aliasing problems.

[2] The Head sequence is provided by University of Tsukuba.

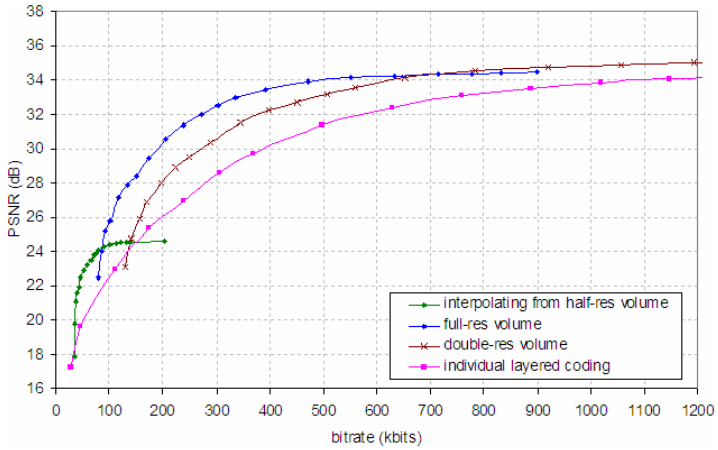


Fig. 15 The performance of the proposed layered coding scheme of the *Head* sequence.

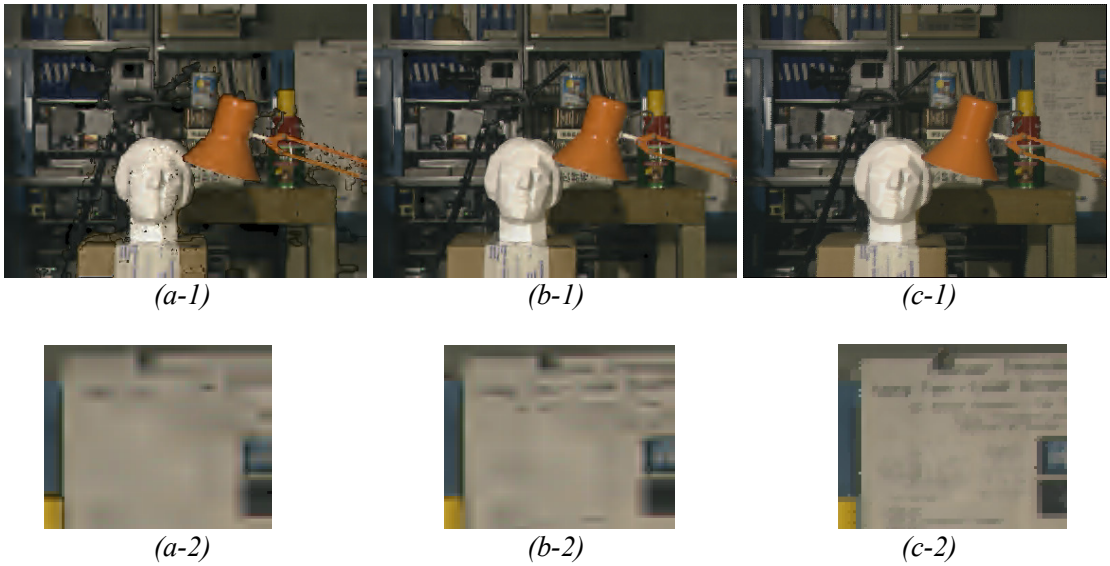


Fig. 16 (1) The reconstructed *Head* images at 35kbits/view rendered from (a) individual layered coding approach, (b) full-resolution volume and (c) half-resolution volume. (2) The magnified texture of the reconstructed images of (1).

5.2 View Synthesis Performance

The view synthesis performance was tested by taking one of the five reference views of the Leo test images off to be used for view synthesis assessment. The volume was constructed using the remaining four views and coded using the proposed schemes. The virtual view was then synthesized with the same geometry of the removed reference view and the image quality was compared to the original view. The average PSNR of the three synthesis qualities are demonstrated with solid lines in Fig. 17. The compression performance is also shown with dash lines in this figure. By comparing to the results presented in Fig. 13, the results presented in Fig. 17 obviously prove that the quality of the volume constructed from five views is better than that constructed from four views. It confirms that the volumetric representation benefits the multi-view image sequences in terms of both modeling accuracy and overall

data compression. Comparing to the compression performance, the view synthesis performance is inferior up to 0.8 dB. However, the view synthesis performance of the proposed coding scheme is higher than that of the traditional scheme.

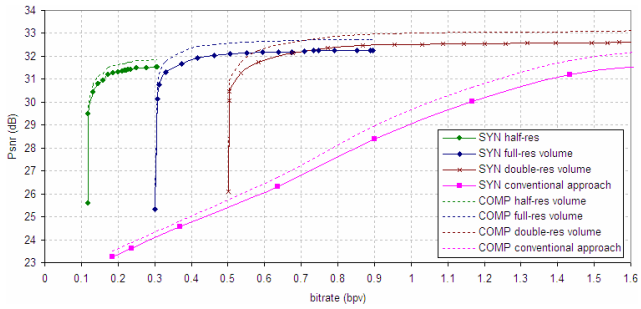


Fig. 17 The view synthesis performance of *Leo* volume from four views

The advantage of the proposed data rearrangement approach over the traditional scheme is that the correlation between each depth plane of the volume of the proposed scheme is increased, whilst the high frequency along the depth plane direction is reduced. Therefore, the wavelet filtering attains more efficiency thereby achieving high compression performance. However, the subjective quality, especially at low bitrates, is not constant around the reconstructed volume. The front view of the reconstructed object achieves higher quality than the side view of the object. This is because the high frequencies containing the side-view details are coarsely coded and truncated to achieve the target bitrate, whilst the low frequencies containing the front-view details are coded with higher proportion of the bitrates. As shown in Fig. 18 (c), some information in the side view of the 300k-reconstructed *Leo* is lost, so the surface of this area is not smooth. However, the reconstructed *Leo* of the non-rearranged data shows a less smooth surface around the object even at higher bitrate as illustrated in Fig. 18 (d). Noticeably, the proposed algorithm significantly outperforms the conventional approach. As shown in the magnified pictures, although the conventional approach is coded at higher bits, Fig. 18 (a-2) shows smoother texture with sharper edges than that of shown in Fig. 18 (d-2).

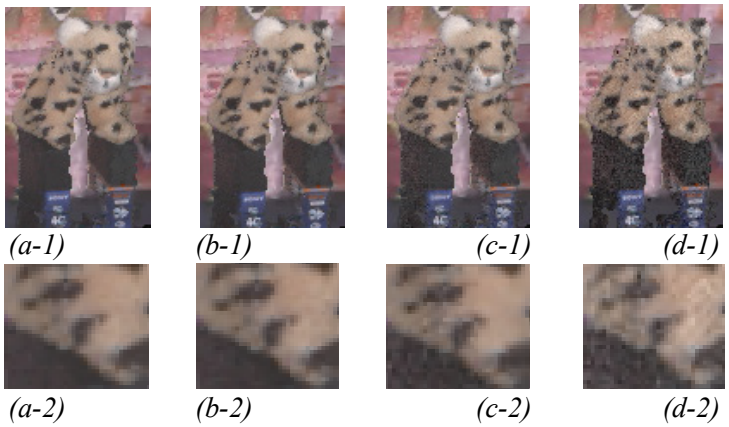


Fig. 18 The synthesized *Leo* images with the proposed rearrangement approaches coded at (a) lossless (535kbits), (b) 430kbits, and (c) 300kbits. (d) The synthesized view without data rearrangement coded at 580kbits. (2) The magnified texture of synthesised views of (1).

6. IMPROVING COMPRESSION PERFORMANCE BY RESIDUAL CODING

Residual coding can be included in the coding technique to improve the quality of the reconstructed images; however, the residuals do not have any contribution to the quality of the view synthesis. In other words, the residual coding improves the compression performance of the codec but may deteriorate the view synthesis performance at a particular bitrate. In this section, the effect of residual coding on the compression and view synthesis performances is investigated.

Volumes constructed utilizing information from five reference views are used for producing the simulation results. The reconstructed images without employing residuals are used for evaluating the view synthesis performance. Open-loop operation is exploited to preserve the scalability feature of the codecs. The residual of each view is the difference between the original image and the direct reconstructed image from the volume without quantization or coding. As a result, the residuals encoded at all bitrates are similar, unlike the residuals in the MPEG standard video codec.

The maximum compression performance is first inspected by varying the bit allocation between the object coding and the residual coding. Fig. 19 illustrates the results for the maximum compression performance and the results for the synthesised views at full resolution with and without including residuals in order to show the possible maximum view synthesis performance. Fig. 19 shows that, the residuals do not greatly improve the quality of the reconstructed image at low bitrate but significantly exacerbate the quality of view synthesis. In contrast, the quality of the reconstructed images is highly improved by adding the residuals, whilst the quality of view synthesis is degraded up to 0.1 dB at high bitrate. Fig. 20 (a-1) and Fig. 20 (a-2) show the compressed Leo images at bitrate of 0.4 and 0.7 bpv respectively when residuals were employed in the compression scheme. Fig. 20 (b-1) and Fig. 20 (b-2) illustrate the synthesised view at 0.4 and 0.7 bpv, respectively. The synthesised view of the Leo image, when residuals were not included in the compression process at 0.4 and 0.7 bpv, are shown in Fig. 20 (c-1) and Fig. 20 (c-2), respectively.

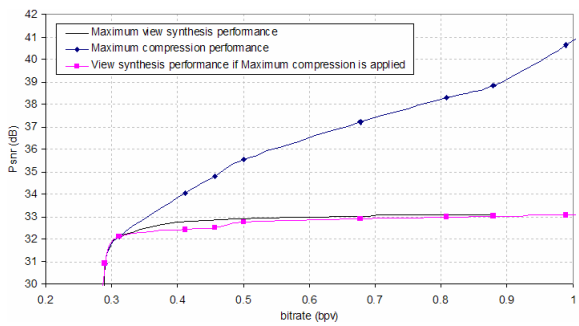


Fig. 19 The highest performance of compression against the performance of view synthesis when including the residual coding

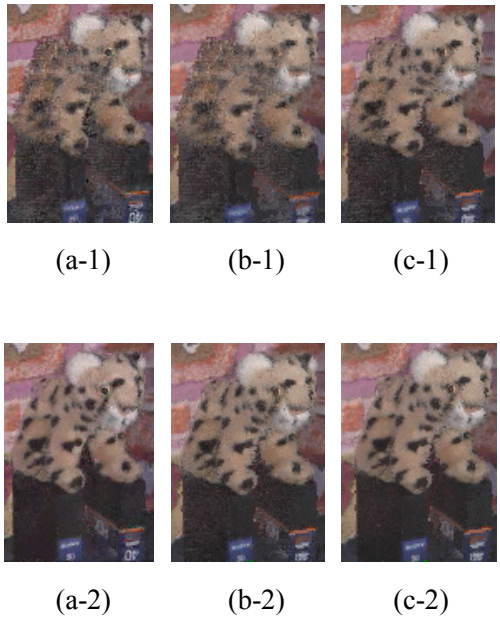


Fig. 20 The subjective results at 0.4 bpv (1) and 0.7 bpv (2). (a) compression performance (apply residuals) (b) view synthesis performance (remove residuals) (c) maximum view synthesis performance (no any residual applied to coding process)

At high bitrate, the quality of the synthesised views approximately saturates to the best achievable quality. A great increase in the number of bits used for coding the volume slightly enhances the quality of the reconstructed images. Therefore, we investigate the compression performance of the proposed technique when the residual coding is only applied at high bitrates where the quality of synthesised views is acceptable. Fig. 21 depicts the coding performance of the proposed algorithm at various acceptable qualities of the synthesised views. Here, the residual coding is employed at bitrates higher than the bitrates in which the quality of the synthesised views reaches the values below the lossless quality at -0.05 dB, -0.1 dB and -0.5 dB. The degradation of -0.5 dB beneath the lossless quality could represent an application in which the views at the original camera positions are significant, whilst the -0.05 dB quality is an example of applications where the good quality of the virtual views is more essential, e.g. 3D rendering, demanding look-around capability and free-view point applications.

This experiment concerns the scalability; hence, the same residuals are coded for all bitrates. At the decoder, the quality of the volume is first selected and then parts of residual bitstream are taken until the target bitrate is obtained. For example, if the acceptable quality of the object is -0.1 dB below the maximum, the decoder consumes 0.53 bpv for the volume. Subsequently, if the total required size is 0.7 bpv, the significant parts of the residuals will be decoded with the remaining bit budget of 0.17 bpv. By using this approach, the decoder can obtain the flexibility of both image size and quality. However, highest compression performance cannot be achieved by choosing the low-quality decoded volume. This is because the residuals are obtained from the original volume but not directly from the reconstructed images at each particular bitrate.

Fig. 22 and Fig. 23 show the subjective results for the maximum achievable quality of the synthesised views at three acceptable qualities; -0.05, -0.1 and -0.5 dB. From these figures, it is obvious that the compression performance is enhanced by adding residuals. From Fig. 23 (d), it is noticeable that an acceptable PSNR value can be achieved at low bitrate. Fig. 23 (d) gains the same PSNR as images in Fig. 23 (b) and (c), but exploits lower bitrates. However, they have different subjective qualities, i.e. the smoother texture is achieved by coding at higher bitrates, whilst a sharper edge is achieved with lower bitrates. It proves that the residuals can correct the errors at edges which comes from the imperfect volume construction or loss of high frequencies due to bitstream truncation.

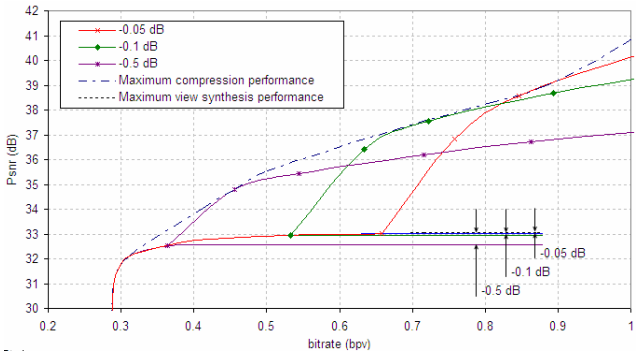
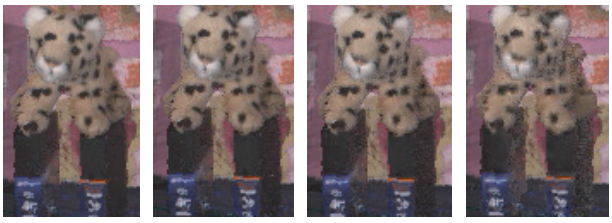
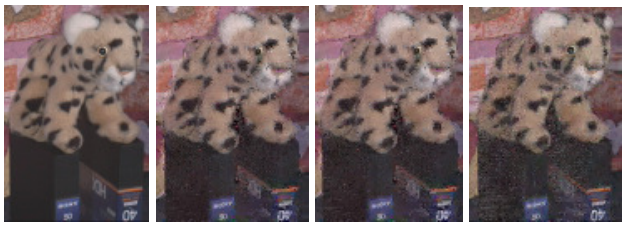


Fig. 21 the compression and view synthesis performance at various acceptable view synthesis qualities.



(a) (b) (c) (d)

Fig. 22 The subjective results of possible maximum quality of view synthesis. (a) best achievable quality (b) -0.05 dB (c) -0.1 dB (d) -0.5 dB.



(a) (b) (c) (d)

Fig. 23 (a) The original view and the reconstructed images with PSNR of 34 dB from criteria of (b) -0.05 dB (0.68 bpv). (c) -0.1 dB (0.56 bpv). (d) -0.5 dB (0.42 bpv).

We also investigate the case that the quality of the volume is selected in the encoder. The decoder will receive the volume of particular quality with the particular residuals for the acceptable quality of the synthesised views. In other

words, the original residuals for each acceptable quality of the synthesised views are different. They are generated from the differences between the reconstructed images at a particular bitrate and the original images thereby achieving the highest compression performance with the price of lossing SNR scalability of the volume. Fig. 24 and Fig. 25 illustrate the objective simulation results and the reconstructed images for the proposed algorithm.

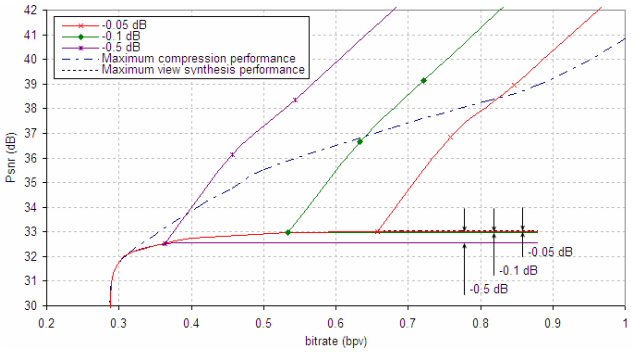
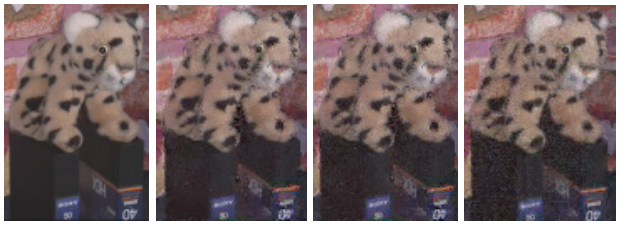


Fig. 24 the highest compression and view synthesis performance at various acceptable view synthesis qualities.



(a) (b) (c) (d)

Fig. 25 (a) The original view and the reconstructed images with quality of 37 dB from criteria of (b) -0.05 dB (0.76 bpv). (c) -0.1 dB (0.64 bpv). (d) -0.5 dB (0.48 bpv).

7. CONCLUSIONS AND FUTURE WORK

This paper proposed a novel approach for colour volumetric coding which is suitable for realistic view synthesis applications. Two algorithms were proposed to rearrange the volumetric data: one uses the characteristics of the human eye sensitivity to diminish the size of the chrominance information and the other reduces the number of components to compact volumetric data. Moreover, the adaptive resolution was introduced to support the scalability and improve the quality of the large image as well as the quality of the view synthesis. Additionally, the effect of residual coding on compression and view synthesis was investigated. Clearly, the residuals can improve the compression performance but may deteriorate the view synthesis performance, especially at low bitrates.

The proposed approaches showed great improvement in the wavelet coding environment and possibly enhance the performance of other coding methods, such as H.264. Moreover, the proposed schemes can possibly be applied to 4D data which represents a time-varying volumetric imaging system.

8. REFERENCES

[1] F.F. Rodler, "Wavelet based 3D compression with fast random access for very large volume data," *IEEE Proc. 7th Pacific Conf. on Computer Graphics and Applications*, pp. 108-117, 1999.

[2] Z. Xiong, X. Wu, S. Cheng and J. Hua, "Lossy-to-Lossless Compression of Medical Volumetric Data Using Three-Dimentional Integer Wavelet Transforms," *IEEE Trans.on Medical Imaging*, vol.22, pp. 459-470, 2003.

[3] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto, J. Cornelis, "Wavelet coding of volumetric medical datasets," *IEEE Trans. on Medical Imaging*, vol.22, pp.441-458, 2003.

[4] Y. Gao and H. Radha, "Multi-View Image Coding Using 3-D Voxel Models," *IEEE Proc. ICIP'05*, vol.2 , pp. 257-260, 2005.

[5] M. Deering, "Geometry Compresion," *ACM proc. Comput. Graph.*, pp.13-20, 1995.

[6] G. Taubin and J. Rossignac, "Geometry Compression through Topological Surgery," *ACM Trans. Graph.*, pp. 84–115, 1998.

[7] A. Jeong-Hwan, K. Chang-Su, H. Yo-Sung, "Predictive Compression of Geometry, Color and Normal Data of 3-D Mesh Models," *IEEE Trans. on Circuits and Systems for Video Technologies,* vol.16, pp. 291- 299, 2006.

[8] K. Chang-Su, S. Lee, "Compact Encoding of 3-D Voxel Surfaces based on Pattern Code Representation," *IEEE Trans. on Image Processing*, vol.11, pp. 932 – 943, 2002.

[9] M. Magnor, P. Ramanathan, B. Girod, "Multi-View Coding for Image-Based Rendering Using 3-D Scene Geometry," *IEEE Trans. on Circuits and Systems for Video Technologies,* vol.13, pp. 1092- 1106, 2003.

[10] "Information Technology - JPEG 2000 Image Coding System: Extensions," ISO/IEC JTC1/SC29/WG1, 15444-2, May 2004.

[11] "Information Technology - JPEG 2000 Image Coding System: Core Coding System," ISO/IEC JTC1/SC29/WG1, 15444-1, 2000.

[12] A. Tzannes, " Compression of 3-Dimensional Medical Image Data Using Part 2 of JPEG 2000," Aware, Inc., Nov, 2003.

[13] N. Anantrasirichai, C. Nishan Canagarajah, David W. Redmill and David R. Bull, "Volumetric Representation for Sparse Multi-views," in *IEEE Proc. ICIP*, 2006, pp. 1221-1224.

[14] R. Koch, M. Pollefeys, L. Van Gool, "Robust Calibration and 3D Geometric Modeling from Large Collections of Uncalibrated Images," in *Proc. DAGM Pattern Recognition Symp.*, pp. 413-420, 1999.

[15] P. Schelkens, A. Munteanu, A. Tzannes, C. Brislawn, "JPEG2000 Part 10 – Volumetric Data Encoding," *IEEE Proc. ISCAS'06*, pp.3874-3877, 2006.

[16] N. Anantrasirichai, C. Nishan Canagarajah, David W. Redmill and David R. Bull, "Dynamic Programming for Multi-view Disparity/Depth Estimation," *IEEE Proc. ICASSP'06*, vol.2, pp.269-272, 2006.