DISPLAY DEPENDENT CODING FOR 3D VIDEO ON AUTOMULTISCOPIC DISPLAYS

Vikas Ramachandra, Matthias Zwicker and Truong Q. Nguyen

CALIT2, University of California, San Diego, CA 92093

ABSTRACT

A method to adaptively code multiview videos has been proposed which uses the depth characteristics of automultiscopic multiview displays. It is found that for the 3D scene seen on multiview displays, regions appearing at large depths are rendered blurry. The proposed method identifies such regions and uses fewer bits to code them. Also, greater number of bits are used for regions which appear sharp on the 3D displays. The overall quality is better than regular AVC/H.264. For compatibility with the framework of scalable multiview coding, we introduce *depth scalability*. This ensures that the (hierarchical layerwise) encoded video bitstream can be optimally displayed on various displays with different depth properties.

Index Terms— Multiview displays, depth dependent coding.

1. INTRODUCTION

Automultiscopic and autostereoscopic displays show stereoscopic and multiview 3D images respectively, that can be viewed without special glasses. Further, automultiscopic displays incorporate motion parallax, which gives the viewer more freedom, i.e. he does not need to keep his head position fixed. Also, the user can be present anywhere in the reasonably big viewing zone. Recently, there has been an effort to standardize the coding standards of multiview video. These proposals are based on extensions of the MPEG 4 Part 10 a.k.a H.264 a.k.a AVC video coding standard [1],[2],[3],[4]. These methods take advantage of the correlations between the different camera view images, and compute the disparity map. The disparity compensated error frame is coded using regular or vanilla H.264 method, specifically using the same quantization matrix for all regions in the error frame. However, based on a frequency domain analysis of automultiscopic displays, it was shown in [6] that regions of the 3D scene which are beyond the depth of field of the display appear blurry to the viewer. We propose a technique to identify such regions and modify the H.264 coding scheme such that fewer bits are allocated to these regions. At the same time, regions which appear sharp on the display are the ones which suffer the most due to the quantization of the vanilla H.264. The proposed method addresses this drawback of vanilla H.264 by allocating more bits (i.e. finer quantization) to such regions. In other words, there is a reallocation of bits adaptively based on the local region depth in the 3D scene being displayed, and the depth characteristics of the 3D display. In practice, over a typical network, an encoder is required to send a video stream which needs to be displayed on different types of 3D displays. In this regard, the proposed method fits into the scalable coding framework, since the encoder can now send a layered video stream; the decoder(s) only use the required number of layers from the common bitstream as dictated by the depth of field capacity of their respective 3D display(s). We call this concept *depth scalability* and is explained in a later section. We proceed with an overview of the multiview extensions of H.264 in section 2, and a frequency space analysis of 3D displays in section 3. The proposed framework for depth adaptive coding is illustrated in section 4. In section 5, we outline how this method can be used practically with simultaneous broadcast to multiple types of 3D displays. Section 6 presents the results followed by the conclusion in section 7.

2. MULTIVIEW EXTENSIONS OF H.264

Recently, there have been proposals for extending H.264 for multiview video [1],[2],[3], [4]. The common feature among these proposals is disparity compensated predictive coding. These efforts have been jointly called multiview video coding (MVC). In MVC, one of the views is chosen as the reference and is first coded as an intra frame, or (temporal) motion compensated predictive coded with respect to the same view images at other time instants. Then, using the reference view(s), each of the other views are disparity compensated. Then, the error frame between the disparity compensated and actual views is calculated, and coded using H.264. Different coding modes can be used for different macroblocks in the error frame similar to standard video coding, for example, motion compensated + residual coding, motion compensation with or without residual coding, etc. The decisions for the coding modes for macroblocks in different regions of the views do not take into account the display properties. The proposed method tries to incorporate knowledge of the characteristics of common 3D displays into the coding pipeline. For this, one needs to understand the interaction of the 3D displays on the scene depth properties, which is explained in

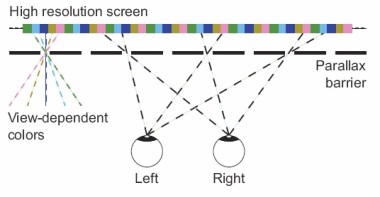

Fig. 1. The automultiscopic display

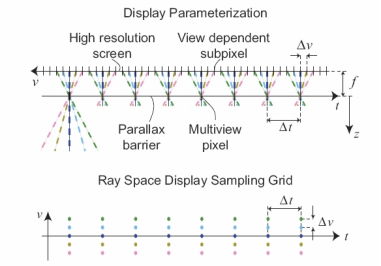

Fig. 2. The ray-space characterization of the display

the next section.

3. FREQUENCY DOMAIN ANALYSIS OF 3D DISPLAYS

Figure 1 shows a parallax barrier based automultiscopic display. The barrier ensures that the left and right eye see different views which creates a 3D i.e. depth perception. Due to the use of multiple views in the display, the user sees new views for small angles of head tilt/rotation resulting in a realistic experience.

The ray space representation was introduced in [5] and extended in [6]. Automultiscopic displays seek to reproduce the light array for every location and direction in the viewing zone. We parameterize light rays by their intersection with two planes. For a parallax-barrier display, we use the parallax barrier plane as the t coordinate, and the high resolution screen as the v coordinate (Figure 2). We follow [5]'s conventions that the t and v axes have opposite orientations and the v coordinate of a ray is relative to its t coordinate. All rays intersecting the t plane at one location correspond to one multiview pixel, and each intersection with the v plane is a view-dependent subpixel. We call the number of multi-view pixels the spatial resolution and the number of view-dependent subpixels per multi-view pixel the angular resolution. The display rays form a higher dimensional grid in ray space. Each ray in the top of Figure 2 corresponds to one sample point at the bottom of the figure. Current digital automultiscopic displays provide only horizontal parallax, i.e., they sample only in the horizontal direction on the v plane. Hence, we can treat each scan line on the t plane independently, which leads to a two-dimensional ray space. We use the term display view to denote a slice of ray space with $v = const$ (note that these views are parallel projections of the scene). Without loss of generality, we assume the distance between the planes f has unit length (see Figure 2), and we omit factors f in our equations.

Display bandwidth: The sampling grid in Figure 2 imposes a strict limit on the bandwidth that can be represented by the display, known as the Nyquist limit (Figure 3 left). Let us denote angular and spatial frequencies by ϕ and θ, and sample spacing by Δv and Δt. Then the display bandwidth is given

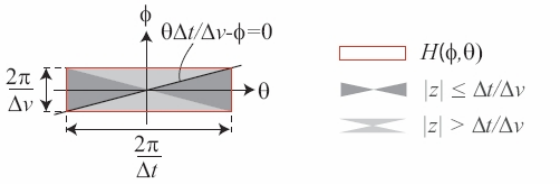

Fig. 3. Depth of field of the display

by

$$H(\phi,\theta) = \begin{cases} 1 \text{ for } |\phi| \le \pi/\Delta v \text{ and } |\theta| \le \pi/\Delta t, \\ 0 \text{ otherwise.} \end{cases} \quad (1)$$

It was shown in [5] that the spectrum of a light field, or ray space signal, of a scene with constant depth is given by a line $\phi/z + \theta = 0$, where z is the distance from the t-plane. From Figure 3, we see that for scenes at depths $|z| \le \Delta t/\Delta v$, their spectral lines intersect the rectangular display bandwidth on its left and right vertical boundary. (The rectangle corresponds to the prefiltering which removes the aliased lightfield components). This means these scenes can be shown at the highest spatial resolution $\theta = \pi/\Delta t$ of the display. However, for scenes with $|z| > \Delta t/\Delta v$, their spectra intersect the display bandwidth on the horizontal boundary. As a consequence, their spatial frequency is reduced to $\theta = \pi/(\Delta v z)$. This is below the spatial resolution of the display, hence these scenes will appear blurry. We call the range $|z| \le \Delta t/\Delta v$ that can be reproduced by a given 3D display at maximum spatial resolution as the "depth of field" of that display. Remaining regions appear blurry and a method to identify them is described in the next section.

4. THE PROPOSED CODING TECHNIQUE

4.1. Identification of blurry regions on the display

In this section, we try to identify regions of any given 3D scene which will not be displayed at their full spatial resolution given a *particular* 3D display. We explain our procedure below for identifying regions which are displayed sharply for the simple case when all the camera positions differ by a

translation along one axis only. (This could be extended to arbitrary camera setups with some more work.)

Form epipolar images: Stack corresponding rows of different view images to form an epipolar image stack. If there are k views, each view image of size $M \times N$, then the epipolar stack will have M epipolar images of size $k \times N$ each.

Edge detection: On each epipolar image, perform standard Canny edge detection which returns a binary edge map corresponding to each epipolar image.

Directional filtering: Every binary map is filtered with a set of directional Gaussian filters of support $l \times l$, each filter is a rotation and shear of a base Gaussian filter (the base filter is aligned along the X-Y axes). Let $-\theta$ to θ be the angular range in the epipolar images. This angular range is swept in (chosen) increments of δ, which means there will be $2\theta/\delta$ directional Gaussian filters.

Flagging the regions with display blur: For each epipolar edge map, the angle/direction ψ corresponding to the best filter response is noted. Let -α to α be the epipolar angular range for which the 3D display considered can render the scene without blur (i.e. this angular region is directly dependent on the 3D display). Now, a new binary map set B is created to identify the sharp and blurry regions for the display (not to be confused with the earlier edge maps). There are k such maps in the set, one for each view. If $-\alpha \leq \psi \leq \alpha$, then that region (i.e. corresponding rows across the different original views) will be rendered sharp on the display, such locations in the map set B is marked with '1'and vice versa.

4.2. Adaptive coding

A standard H.264 codec was modified here, but the rate control and R-D optimization was turned off for easier desired control on our part. Once the regions which will appear sharp and blurry have been demarcated, the map set B is fed into the H.264 codec. During the encoding of every macroblock, the codec is forced to check if the macroblock lies in the sharp regions, in which case it chooses a particular quantization parameter Q_Psharp, and a different quantization parameter Q_Pblur for regions which will appear blurry. In other words, the bits are redistributed such that blurry regions are more heavily quantized (taking advantage of the fact that these regions are anyway not rendered well by the display, hence any distortion is not very visible to the viewer). On the other hand, more bits are allocated to the regions which appear sharp. This is in contrast to the standard H.264 reference codec software, which, due to its fixed quantization parameter (Q_Pfixed) for each frame (in the vanilla case), would have blurred the regions which are rendered sharp by the display, as well as would have unnecessarily allocated more bits to regions which would anyway appear blurry on the display. Instead the proposed adaptive coding takes advantage of the display characteristics for better bit allocation and quantization.

Given a bitrate (i.e. a corresponding Q_Pfixed), and also f, the fraction of pixels in the present image which appear sharp on the display (which can be found from B), we illustrate how to choose Q_Pblur and Q_Psharp. We use the simple rate-distortion modeling as proposed in [7] based on the ρ-domain analysis. After the DCT coefficients are quantized with a quantization parameter q, let ρ be the percentage of zeros among the quantized coefficients. Note that in typical transform coding systems, ρ monotonically increases with q. Hence, there is a one-to-one mapping between them. This implies that, mathematically, rate R and distortion D are also functions of ρ, denoted by $R(\rho)$ and $D(\rho)$. A study of the rate and distortion as functions of ρ is called ρ-domain analysis. It was shown in [7] that:

$$R(\rho) = \theta(1 - \rho) \tag{2}$$

A scheme for estimating the parameter θ (which depends on image content) is outlined in [7]. Let R_{total} be our bit budget. First, we encode the regions which will appear sharp on the display at the desired (lower) Q_Psharp, which is chosen according to the desired quality of the regions which appear in focus. Then we calculate the total number of bits R_{sharp} to encode the sharp regions. Therefore, $R_{blur} = R_{total} - R_{sharp}$ is the available number of bits to encode the regions which appear blurry. Using equation (2), we can estimate the (average) $\rho_{required}$ for the blurry regions. Then, we sweep over the quantization parameters Q_Pblur and quantize the blurry regions, and calculate the ρ_{actual} for each Q_Pblur chosen. We choose the Q_Pblur for which the ρ_{actual} and $\rho_{required}$ match closely. Then, the blurry regions are finally quantized and coded using this value of Q_Pblur.

5. DEPTH SCALABILITY

In the previous sections we assumed that we knew the display depth of field characteristics. However, in practical cases, this information may not be available to the encoder. Moreover, the video might be broadcast over a network, wherein each end user might have a different display. In such a case, it is necessary to have a coded bitstream which is playable on all the different user displays, but at the same time, the encoder must make use of the display characteristics of all the user displays. This fits in very well in a scalable video framework. For this, we introduce a concept called 'Depth Scalability'. The encoder sends a multi-layered bitstream. The lower layers contain the bitstream corresponding to the adaptive coding of the views for the displays with smaller fields of view. The progressively higher layers contain coded enhancements (in terms of the high frequency parts corresponding to better spatial resolution) for the regions which will appear sharper on displays with better fields of view, but not on the displays with smaller fields of view. Thus, the bitstream can cater to multiple (commonly available) displays. At the receiver, based on the type of display, only the required portion of the bitstream

is decoded and used, which corresponds to the depth of field capacity of the particular display. Thus, "depth scalability" is locally adaptive spatial scalability in the scalable video coding framework.

6. RESULTS

In this section, we compare the proposed method to vanilla H.264 for 2 multiview sequences, namely 'Elephant' (Figures 4 (a) and 4 (b)) and 'Waterfall'(Figures 5 (a) and 5 (b)). Please zoom in to see the results clearly. The display we used was a NewSight 32" AD3 automultiscopic display. In the 'Elephant' sequence, the elephant's eye and trunk are within the depth of field of the display, hence the proposed method performs a finer quantization ($Q_P sharp = 37$) on the macroblocks in these regions. This is also true for the face in the 'Waterfall'. For the regions which appear blurry on the display (the waterfall in the 'Waterfall' sequence, the leaves in the 'Elephant' sequence), a coarser quantization ($Q_P blur = 45$) is used, which is calculated as explained before. This is in contrast to the vanilla H.264 codec which uses the same $Q_P = 40$ for all the macroblocks. The same overall bitrate was used for both the proposed as well as the standard H.264 schemes. On the 3D display, the sharp regions look much better when coded with the proposed method, whereas the blurry regions do not look worse than vanilla H.264, because of the blur introduced by the display itself. Also, the blocking artifacts are fewer in the prominent sharp regions when the proposed method is used, as is clearly visible in the face region of the 'Waterfall' image. Here, we reproduce a frame for each sequence (Figures 4 and 5) by taking a screenshot of what is displayed, to get get a visual feel of the proposed method (the 3D effect cannot be fully reproduced on paper here).

(a) Proposed coding (b) Vanilla H.264

Fig. 4. Comparison for Elephant sequence

7. CONCLUSION

A method to adaptively code multiview videos has been proposed based on the depth characteristics of automultiscopic displays. It is known that the 3D scene seen on such displays has regions appearing at large depths to be blurry. The proposed method aims at identifying such regions and using fewer bits for them, while using more bits for regions which appear sharp on the 3D displays. This results in an overall better 3D scene quality than when using vanilla extensions of H.264. Moreover, this fits well in the framework of scalable multiview coding, with the introduction of a new concept which we call *depth scalability*. Future work involves the analysis of the amount of prefiltering required when the views will be compressed. Compression acts as a high frequency suppressor when quantization removes the higher frequencies, which would mean that one could (at least partially) do away with prefiltering to avoid overly blurring the views.

(a) Proposed coding (b) Vanilla H.264

Fig. 5. Comparison for Waterfall sequence

8. REFERENCES

[1] G. Li and Y. He, "A novel multi-view video coding scheme based on H.264," Fourth Pacific Rim Conference on Multimedia, 2003.

[2] M Drose, C Clemens and T Sikora, "Extending Single-View Scalable Video Coding to Multi-View Based on H. 264/AVC," ICIP 2006.

[3] E Martinian, A Behrens, J Xin, A Vetro and H Sun, "Extensions of H.264/AVC for multiview video compression," ICIP 2006.

[4] P Merkle, K Muller, A Smolic and T Wiegand, "Efficient compression of multi-view video exploiting inter-view dependencies based on H.264/MPEG4-AVC," ICME 2006.

[5] JX Chai, X Tong, SC Chan and HY Shum, "Plenoptic sampling," ACM SIGGRAPH, 2000.

[6] M Zwicker, W Matusik, F Durand and H Pfister, "Antialiasing for automultiscopic 3D displays," ACM SIGGRAPH, 2006.

[7] Z He and SK Mitra, "A linear source model and a unified rate control algorithm for DCT video coding," IEEE Trans. on Circuits and Systems for Video Technology, 2002.