SHAPE ADAPTATION FOR LIGHT FIELD COMPRESSION

Chuo-Ling Chang, Xiaoqing Zhu, Prashant Ramanathan, and Bernd Girod

Information Systems Laboratory, Department of Electrical Engineering
Stanford University
{chuoling,zhuxq,pramanat,bgirod}@Stanford.EDU

ABSTRACT

We propose a method of using shape adaptation for compression of light fields of 3-D objects. Shape adaptation is incorporated into two light field coders, both applying disparity compensation with an explicit geometry model, to improve the compression efficiency. The shape information can be derived at the decoder from an accurate geometry model. If the available geometry is inaccurate, we propose to code the exact 2-D shapes with the aid of the approximate geometry. Experiments show that shape adaptation greatly improves the compression performance of both light field coders.

1. INTRODUCTION

Image-based rendering has become an important graphics technique, especially for interactive photo-realistic applications. A *light field* [1][2] is a data set for image-based rendering. It captures the outgoing radiance from a particular scene or object, at all points in 3-D space and in all directions. In practice, the light field can be represented as a 2-D set of 2-D camera views. A novel view from an arbitrary position and direction can be generated by appropriately combining image pixels from the acquired views.

The uncompressed size for a large photo-realistic light field can easily exceed tens of Gigabytes. Compression is therefore an important component of any light field system. A commonly used technique for light field compression is disparity compensation, analogous to motion compensation in video compression, which aligns neighboring views for effective exploitation of the coherence among them.

For predictive coding approaches with disparity compensation, disparity values are either specified for a block of pixels or inferred from a geometry model [3][4]. The views are predicted, via the disparity values, from previously encoded reference views, and the residuals are encoded. Another way to achieve disparity compensation is to re-parameterize the views to an aligned structure, for instance, a set of view-dependent texture-maps [5] or the surface light field [6], using an explicit geometry model.

For light fields depicting an object, only the object itself is of interest and the background is not needed. For compression of such data sets, current predictive coding approaches cannot entirely omit the background. Moreover, the discontinuities at object boundaries are difficult to predict and expensive to code. The re-parameterization approaches can alleviate these problems, since they only code the part of the views covered by the geometry model. For the same reason, however, a geometry model with high accuracy is required, otherwise the object shape cannot be recovered exactly.

Similar problems have been observed in image and video compression, and shape adaptation has been used there to mitigate such problems. The shape-adaptive discrete cosine transform (SA-DCT) [7][8] and shape-adaptive discrete wavelet transform (SA-DWT) [9], for instance, have gained popularity due to recent work in object-based video coding, incorporated into standardization efforts such as MPEG-4.

In this paper, we propose a method of using shape adaptation for significantly more efficient compression of light fields of 3-D objects. Specifically, we incorporate shape adaptation into two light field coders, both of which perform disparity compensation using an explicit geometry model. The 2-D shape of the object in each view can be obtained from the projection of the geometry. For the case that the geometry model at hand is not accurate enough to match the exact 2-D shapes, we propose a shape coding technique to predict from approximate geometry.

The remainder of the paper is organized as follows: In Section 2, we describe the two light field coders that we later consider for shape adaptation. In Section 3, we describe the general technique for shape adaptation, and show how these can be applied specifically to two coders. We illustrate the shape coding method when using approximate geometry in Section 4. Experimental results are presented in Section 5.

2. LIGHT FIELD CODERS

In this section we briefly describe the two light field coders to be investigated with shape adaptation later. They share similar structures: disparity compensation for exploiting the coherence among different views, image transforms within each view for exploiting similarities among neighboring pixels, followed by quantization and coding of the coefficients. The techniques adopted by each coder are described in the following sub-sections, and for convenience we shall refer to one scheme as the DCT coder and the other as the DWT coder for the rest of the paper.

2.1. DCT Coder

The DCT coder [10] uses a hierarchical predictive structure to encode all the views. It divides each image into blocks of 8×8 pixels. For each block there are three possible modes of coding: *INTRA*, image coding without prediction; *GEO*, disparity compensation from a geometry model followed by residual error coding; and *COPY*, copying from the block at the same position in a designated reference image.

To encode each block with *INTRA* and *GEO* mode, an 8×8 DCT transform is applied, followed by quantization and run-level-coding of the coefficients. Mode selection is based on minimizing rate-distortion Lagrangian cost function for each block.

2.2. DWT Coder

The DWT coder [11] incorporates disparity compensation into a wavelet transform of the light field data. The first stage of the coder is an *inter-view* transform, where geometry-based disparity compensation is combined with lifting to exploit coherence between different views. The low-pass and high-pass subband images roughly correspond to the average and difference between the views, respectively, after warping to a common view-point.

Each subband image is further decomposed with a multi-level wavelet kernel to exploit coherence among neighboring pixels. Afterwards, the four-dimensionally decomposed wavelet coefficients are encoded by a modified version of the SPIHT coder [12], which re-groups the coefficients into individual blocks for coding and memory efficiency. Finally the bitstreams from all coefficient blocks are truncated and assembled together in a rate-distortion optimized fashion using the Lagrangian multiplier technique.

3. SHAPE ADAPTATION

When the light field of interest is an object, the constituent images contain extraneous background pixels and discontinuities at the object boundaries. In the former, we encode unnecessary pixels. In the latter, there is increased energy in the high frequency components. In both cases, this leads to inefficiency in the coding.

We therefore propose to mitigate these two effects by shape adaptation, which can be applied to the different stages of light field compression as explained in the following sub-sections. In Section 4 we will describe how the 2-D object shape at each view is obtained and coded.

Note that by setting the object shape to the entire image, the shape adaptation technique reduces to conventional coding without shape adaptation, therefore no restriction on scene content is introduced by the scheme.

3.1. Improved Disparity Compensation

When shape information is available, better disparity compensation at the object boundaries can be performed. In particular, with an inaccurate geometry model, an object pixel in one view may be disparity-compensated to the background in another view. The contrast between object and background pixels may give rise to large residual errors that are expensive to code. With knowledge of exact object boundaries, on the other hand, the prediction can be obtained from the nearest object pixel instead of the background.

Another problem with disparity compensation from inaccurate geometry is that some object pixels near the boundary do not have a corresponding point on the geometry. In this case disparity compensation is not applicable, resulting in large residual errors at the edges. With shape information, all object pixels are easily identified. The disparity values of those pixels unaccounted for by the geometry can be simply extrapolated from neighboring pixels, thus all the object pixels can be disparity-compensated properly.

It is worth noting that shape adaptation for disparity compensation is applicable to both coders. In the actual implementation, however, only the DWT coder has incorporated the procedures mentioned above.

3.2. Shape-Adaptive Transform and Coefficient Coding

In the original schemes without shape adaptation, the DCT coder typically invokes the *COPY* mode to encode the blocks containing constant background, and the DWT coder allocates no bits to a coefficient block of all zeros. In both cases, however, the overhead for side information cannot be omitted, and more importantly, the boundary regions containing both background and object pixels cannot be handled easily. As a result, a significant portion of the bitstream is spent on encoding the background.

Alternatively, with shape information, a shape-adaptive transform can be applied to each view in lieu of the conventional transform. Combined with modified coefficient coding procedures, the background can be skipped entirely. In addition, since the shape-adaptive scheme avoids performing the transform across object boundaries, extraneous high frequency components are avoided, contributing to improved coding efficiency and enhanced reconstruction quality.

For the DCT coder, if the block lies completely outside the object boundary, it is simply discarded; if it resides completely within the object, it is coded conventionally, i.e., with 8×8 DCT. Only when the block contains both object and background pixels, the Shape-Adaptive DCT (SA-DCT) [7][8] is invoked to produce the same number of coefficients as that of the object pixels in the block. The subsequent run-level-coder is modified to encode only these coefficients.

For the DWT coder, the Shape-Adaptive DWT (SA-DWT) as proposed in [9] is adopted. For each view the transform is performed on the entire image, generating as many wavelet coefficients as object pixels. The SPIHT algorithm is modified to disregard irrelevant zero-trees. Note that conventionally bitstreams from SPIHT coding are further compressed by a context-based adaptive arithmetic coder, whereas with shape adaptation, there will likely be a much smaller performance gain from appending the arithmetic coder [13]. Therefore, the need of arithmetic coding is eliminated, and coding complexity can be reduced without much sacrifice in compression efficiency.

4. SHAPE CODING

The 2-D shape of the object in a particular view can be obtained by image segmentation techniques. For the proposed shape-adaptive light field compression scheme, both the shape in each view and the geometry model need to be available at the decoder. If the projection of the geometry is consistent with the 2-D object shape in each view, however, the geometry model itself can account for the shape information. No extra shape coding is needed.

Conversely, if the geometry model is just an approximation, it can only provide approximate shape information. An example of the exact shape, denoted by S, and the approximate shape, denoted by $\hat{S}$, is illustrated in Fig. 1(a)(b).

In such a case, we propose to efficiently code the exact shape as side information, using the approximate shape available. $\hat{S}$ is dilated for several iterations until the white pixels in the dilated image cover all of the white pixels in S. We denote the dilated $\hat{S}$ by $\tilde{S}$, illustrated in Fig. 1(c), which is actually used to predict S. The purpose of the dilation is to ensure that S can always get reference values to be predicted from. The number of dilations is transmitted, and the decoder, knowing this number, can recover $\tilde{S}$ since the geometry is also available.

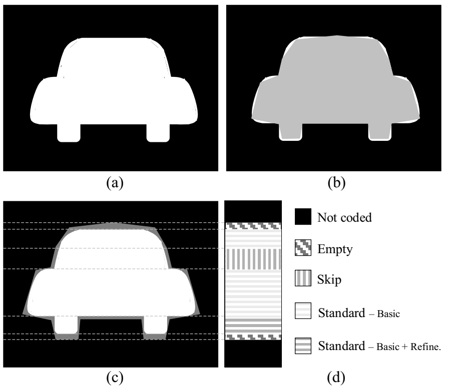


Fig. 1. Shape coding example: (a) Exact shape S showing a front view of a car. (b) Approximate shape $\hat{S}$ from a geometry model, colored in light gray and superimposed on S. (c) Dilated approximate shape $\tilde{S}$, colored in dark gray with S superimposed on it. The dashed lines divide the view into regions with different mode selections. (d) Mode selection of each line: For example, the lines in the region containing the tires uses *Standard* mode with both *Basic* and *Refinement* passes.

$\tilde{S}$, $\hat{S}$ and S are compared for the horizontal scan lines containing white pixels in $\tilde{S}$. We denote a line in $\tilde{S}$ by $\tilde{L}$, the corresponding line in $\hat{S}$ by $\hat{L}$ and that in S by L. Three modes are defined for all possible situations:

- *Empty* mode: This mode is selected when L contains no white pixels, i.e., the object is not contained in this line.
- *Skip* mode: This mode is selected when L contains white pixels and it is identical to $\hat{L}$, i.e., the approximate shape is already identical to the exact shape.
- *Standard* mode: This mode is selected when L contains white pixels but is not identical to $\hat{L}$. In this case, $\tilde{L}$ is used to predict L. Two passes are further defined for this mode:
  - *Basic* pass: The distance between the leftmost white pixel in L and that in $\tilde{L}$ is recorded, followed by the distance between the rightmost white pixel in L and that in $\tilde{L}$. The two distances are predictively encoded using the corresponding distance from the line above.
  - *Refinement* pass: If the line contains more than one run of white pixels in L, it is signaled in the refinement pass. Starting from the leftmost white pixel, length of each run in L is recorded, alternating between white runs and black runs. Note that the length of the last white run does not need to be encoded since it can be derived from the *Basic* pass.

The mode selection of each line for the given example is shown in Fig. 1(d). The mode selections and the associated distances and run-lengths are combined to form a set of symbols, which are further encoded using an adaptive arithmetic coder. Note that as a more accurate geometry is used, the *Skip* mode is selected more often, hence lower bit-rate is needed for shape coding.

5. EXPERIMENTAL RESULTS

Experimental results are shown for two light field data sets. *Buddha* contains 280 views, each with an image resolution of 512×512 pixels, and *Garfield* has 256 views with a resolution of 384×288 pixels. Examples of the views are shown in Fig. 3(a) and Fig. 4(a). The experiments are performed on (Y, C_b, C_r) color representation, with C_b and C_r down-sampled by a factor of 2 in each image dimension. The compression results are shown as the rate-PSNR curves, where the rate is expressed as bit-per-pixel (bpp) accounting for all three color components. The PSNR is computed by averaging over the pixels in the luminance component, including the background.

For *Buddha*, a geometry model with 6126 triangles is available and is encoded at 0.0061 bpp. For *Garfield*, a geometry model with 2048 triangles is reconstructed from the views, and encoded at 0.0015 bpp. The geometry for *Buddha* provides the exact shape information, hence no extra shape coding is needed. Whereas for *Garfield*, the 2-D shapes are coded using the proposed method at 0.0077 bpp, less than half of the bit-rate, 0.0174 bpp, when directly applying *JBIG* [14]. Note that the ratio of the number of object pixels to the total number of pixels is 49.37% in *Buddha* and 16.70% in *Garfield*. The effect of shape adaptation for the two coders are shown in Fig. 2.

For *Buddha*, shape adaptation reduces 10-30% of the bit-rate for the DCT coder, and 20-25% for the DWT coder, to achieve the same PSNR value. Equivalently, it improves the reconstruction quality by 1-2 dB for both coders. For *Garfield*, a 40-50% bit-rate reduction can be achieved for the DCT coder, and a 60-70% re-

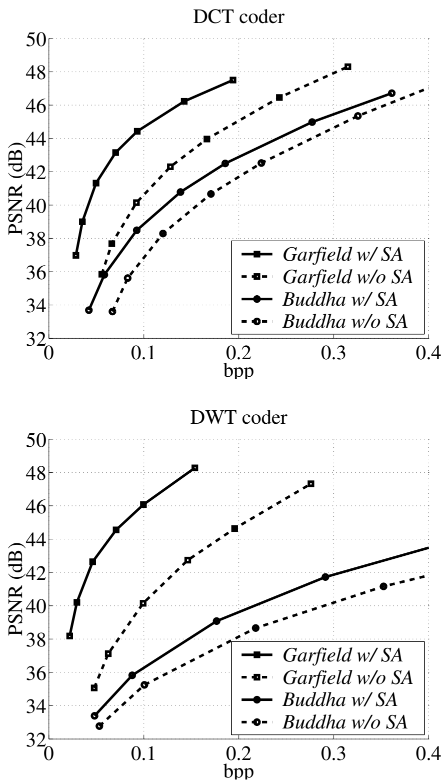


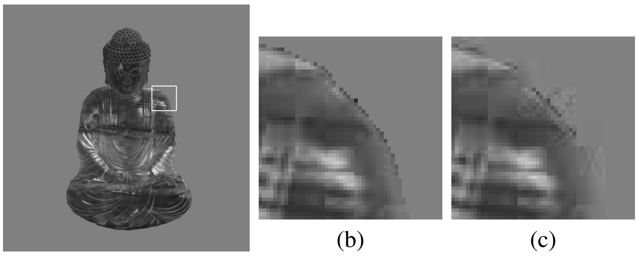
Fig. 2. Effect of shape adaptation (top) the DCT coder (bottom) the DWT coder

duction is observed for the DWT coder. At the same bit-rate, we observe a gain of 3-6 dB in PSNR. The gain for *Garfield* is significantly larger to that of *Buddha*. This can be accounted by the larger ratio of background area in the data set, as well as more pixels at the object boundaries. More importantly, the exact 2-D shape information greatly improves the disparity compensation provided by the approximate geometry model used in *Garfield*.

Examples of the reconstruction from the scheme with and without shape adaptation are shown in Fig. 3(b)(c) for the DCT coder, and Fig. 4(b)(c) for the DWT coder. When coding at similar bit-rates, shape adaptation provides a more faithful representation of the original views, consistent with the PSNR values in Fig. 2. Furthermore, with shape adaptation the object shape is well preserved, whereas in the other case the object boundaries appear to be blurred and distorted.

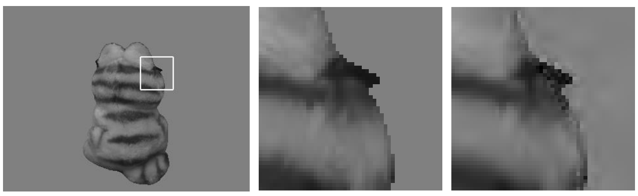
6. CONCLUSIONS

We incorporate shape adaptation for compression of light fields of an object. The use of the 2-D shape information provides better disparity compensation, omits the unneeded background, and avoids coding the discontinuity at object boundaries. The object shape at each view can be directly obtained without extra coding overhead, if an accurate geometry model is available. If the geometry model is only approximate, the exact shape can be efficiently coded using the approximate geometry. This ensures that the object shape is preserved in the reconstructed views. Experiments show that the exact shape information greatly improves the compression efficiency as well as the visual quality of the reconstruction. We observe a reduction in bit-rate for 10-70%, or an improvement in image quality for 1-6 dB, depending on the data set and light field coding method used.



(a) (b) (c)

Fig. 3. Luminance component of *Buddha*: (a) original view with the white box labelling the area of the reconstruction magnified in (b) and (c); (b) DCT coder with shape adaptation: reconstruction at 0.1491 bpp; (c) DCT coder without shape adaptation: reconstruction at 0.1692 bpp (bit-rates include all color components)



(a) (b) (c)

Fig. 4. Luminance component of *Garfield*: (a) original view with the white box labelling the area of the reconstruction magnified in (b) and (c); (b) DWT coder with shape adaptation: reconstruction at 0.0935 bpp; (c) DWT coder without shape adaptation: reconstruction at 0.0932 bpp (bit-rates include all color components)

7. REFERENCES

[1] M. Levoy and P. Hanrahan, "Light field rendering," in *Computer Graphics (Proceedings SIGGRAPH 96)*, August 1996, pp. 31–42.

[2] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Computer Graphics (Proceedings SIGGRAPH 96)*, August 1996, pp. 43–54.

[3] X. Tong and R. M. Gray, "Coding of multi-view images for immersive viewing," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP 2000*, Istanbul, Turkey, June 2000, vol. 4, pp. 1879–1882.

[4] M. Magnor, P. Eisert, and B. Girod, "Model-aided coding of multi-viewpoint image data," in *Proceedings of the IEEE International Conference on Image Processing ICIP-2000*, Vancouver, Canada, September 2000, vol. 2, pp. 919–922.

[5] M. Magnor and B. Girod, "Model-based coding of multi-viewpoint imagery," in *Proceedings SPIE Visual Communications and Image Processing VCIP-2000*, Perth, Australia, June 2000, vol. 1, pp. 14–22.

[6] G. Miller, S. Rubin, and D. Ponceleon, "Lazy decompression of surface light fields for precomputed global illumination," in *Rendering Techniques '98 (Proc. Eurographics Workship on Rendering)*, June 1998, pp. 281–292.

[7] T. Sikora and B. Makai, "Shape-adaptive dct for generic coding of video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 1, pp. 59–62, February 1995.

[8] P. Kauff and K. Schüür, "A shape-adaptive DCT with block-based DC separation and Delta-DC correction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 3, pp. 237–242, June 1998.

[9] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 725–743, August 2000.

[10] P. Ramanathan, E. Steinbach, P. Eisert, and B. Girod, "Geometry refinement for light field compression," Rochester, NY, USA, September 2002, vol. 2, pp. 225–228.

[11] B. Girod, C.-L. Chang, P. Ramanathan, and X. Zhu, "Light field compression using disparity-compensated lifting," in *(submitted) IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-2003), Hong Kong*, 2003.

[12] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on Set Partitioning in Hierarchical Trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.

[13] G. Minami, Z. Xiong, A. Wang, and S. Mehrotra, "3-D wavelet coding of video with arbitrary regions of support," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 1063–1068, September 2001.

[14] ITU-T, "ISO/IEC 11544:1993, Progressive Bi-level Image Compression," 1993.