Proc. Vision, Modeling, and Visualization VMV-2000
Saarbrücken, Germany, pp. 199-203, November 2000

Progressive Compression and Rendering of Light Fields

Marcus Magnor, Andreas Endmann
Telecommunications Laboratory
University Erlangen-Nuremberg
91058 Erlangen, Germany
magnor@lnt.de

Bernd Girod
Informations Systems Laboratory
Stanford University
Stanford, CA 94305-9510
girod@ee.stanford.edu

Abstract

The paper presents a progressive light-field coding scheme based on four-dimensional wavelet decomposition. At 100:1 compression, reconstructed light-field images are visually indistinguishable from the original images, enabling even large light-field data sets to fit into local memory. Progressive decoding allows continuous adjustment of rendering quality to available computational resources. The proposed scheme is validated using two well-known light fields. Rendered image quality remains acceptable even if less than one thousandth of the original light-field information is decoded.

1 Introduction

Photorealistic rendering results as well as scene-independent rendering performance are very attractive advantages of Light Field Rendering (LFR) over conventional geometry-based rendering techniques. Since LFR was proposed in 1996 [1, 2], however, two fundamental obstacles to widespread use of LFR remain unsolved: tremendous amounts of image data need to be stored and transmitted to achieve convincing rendering results, and software-based LFR is computationally very demanding, currently restricting interactive LFR to high-end graphics workstations.

To overcome the former problem, a number of coding techniques have been applied to light-field compression. Vector quantization features very fast decoding of random image segments, enabling interactive decoding and rendering rates. Coding efficiency, on the other hand, is low, yielding less than 35:1 compression at moderate reconstruction results [3]. Compression ratios up to 200:1 are attainable only if vector quantization is followed by entropy coding, sacrificing fast access to arbitrary data segments and thus losing interactivity [1].

Several transform coding techniques have been applied to light-field compression. At significantly higher decoding complexity than vector quantization, the block-based discrete cosine transform, adopted from still-image compression, achieves comparably low 20:1 compression [4], as do spherical harmonic functions [5]. Recent investigations into wavelet-coding of light fields [6, 7] show promise with regard to time-critical decoding, yet so far compression factors of only $\approx 20:1$ have been achieved.

Because light fields depict static 3-D scenes, much higher compression is attainable if similarities between light-field images are exploited. Compression ratios exceeding 1000:1 are obtained if disparity-compensating coding schemes are employed [8, 9]. High compression is desireable, e.g., for light-field transmission over the Internet. Exploitation of inter-image similarities, however, introduces multiple dependencies among light-field images, preventing immediate access to arbitrary data segments during decoding.

The wavelet-based coding scheme presented in this paper offers a solution to both problems: the codec achieves high compression to allow even large light fields to fit into local memory, and, in addition, it also features progressive decoding and rendering which enables trading rendering speed vs. image quality, making possible continuous adjustments to changing computational resources. Fast random access to arbitrary data segments is ensured by doing without image prediction, avoiding dependencies among light-field images.

The paper is organized as follows. In the next section, we describe the wavelet-based decomposition of the light field into frequency subbands. We go on to explain how wavelet coefficients are efficiently coded in order of importance. After describing progressive decoding and rendering, the scheme is validated using the publicly available *Buddha* and *Dragon* light fields.

2 4-Dimensional Wavelet Decomposition

As light fields are composed of multiple images, standard still-image coding techniques can directly be employed to light-field compression, yielding comparable compression ratios [1, 4]. Wavelet-based image coding offers an attractive alternative to common DCT or vector quantization schemes by featuring image reconstruction at different levels of detail [10, 11]. For light-field compression, much higher coding efficiency can be expected by exploiting pixel correlation not only within single images but also between different light-field images. Wavelet-based image coding can be straightforwardly extended to higher-dimensional data structures: a 4-D wavelet-based light-field coding scheme is presented, featuring high compression as well as progressive decoding.

Figure 1: Composition of pixels from all 32×32 images of the *Dragon* light field at coordinate $s, t = 128$: pixel colors vary gently, indicating high correlation along the U and V direction.

In the following, we adopt the nomenclature used in [1], with u and v specifying an image of the array, while s and t denote pixel position within an image. U, V, S and T are then mutually orthogonal directions, expressing the light field's four-dimensional data structure.

The light field exhibits statistical properties very similar to conventional 2-D images. Adjacent light-field pixels are highly correlated not only within single light-field images (dimensions S, T), but also between neighboring images (U, V), Fig. 1. To exploit pixel correlation in all 4 directions, the light field is wavelet-transformed along the S, T, U and V dimension separately applying a one-dimensional wavelet kernel adopted from [11] with a support of 8. The light field is hierarchically decomposited into frequency subbands, Fig. 2. The obtained four-dimensional wavelet coefficient array constitutes a multi-resolution representation of the light field. During decoding and rendering, the displayed level of detail can be adjusted by back-transforming only low-frequency wavelet coefficients. The number of wavelet trans-

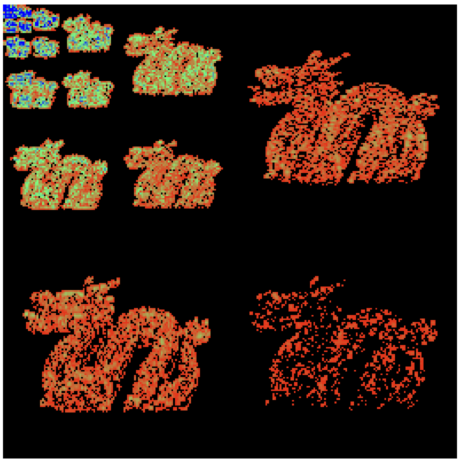


Figure 2: 2-dimensional subband decomposition of a single *Dragon* light-field image (decomposition along S,T). The wavelet-coefficient array resembles a multi-resolution pyramid, with lowest image resolution in the upper left corner. Similarities between different subbands are obvious, and coefficient energy decreases from low to high spatial frequencies. These general properties are exploited to efficiently code the wavelet coefficient array in order of descending coefficient magnitude. For light-field coding, the data is subband-decomposed also along the U and V directions. The resulting 4-D coefficient array exhibits the same general properties as the depicted 2-D case.

forms is further reduced if only coefficients contributing to the rendered view are regarded.

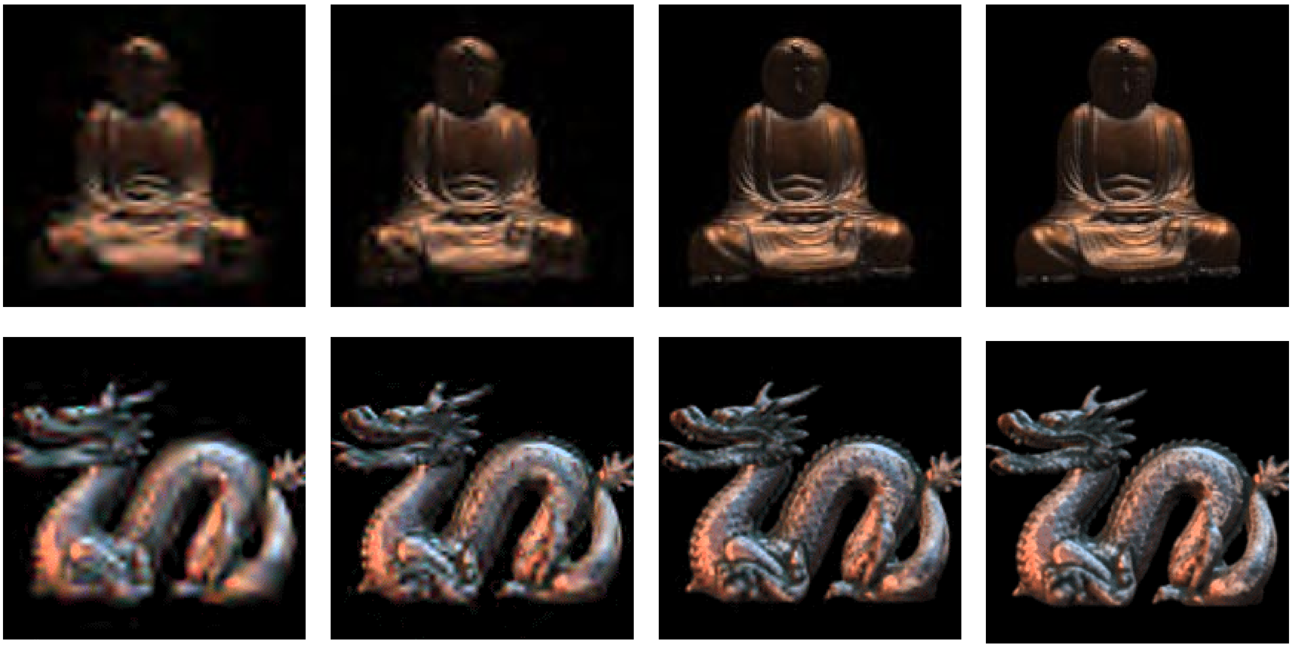
For storage and transmission, the 4-D wavelet representation of the light field must be compressed. The *Set Partitioning in Hierarchical Trees* (SPIHT) algorithm [11] offers an efficient scheme for coding wavelet coefficients, developed for conventional 2-D still-image compression. Compression rates are comparable to other state-of-the-art coding techniques. The SPIHT algorithm codes wavelet coefficients in order of importance: large-magnitude coefficients are coded early on, while small coefficient values are regarded later in the bitstream. Reconstructed coefficient values are first coarsely initialized and gradually refined as bitstream decoding continues. Correlations between wavelet coefficients in different frequency subbands are exploited to code the positional arrangement of the magnitude-ordered coefficients.

We extend the SPIHT algorithm to be capable of coding the light field's four-dimensional wavelet representation, exploiting coefficient correlations among subbands along all 4 directions. The wavelet coefficients are coded progressively and in descending order of magnitude, preserving in our new 4D-Wavelet codec the conventional SPIHT-algorithm's advantages.

3 Progressive Decoding

For rendering, the bitstream is loaded into local memory. While the bitstream is sequentially decoded, only wavelet coefficients contributing to the rendered image are reconstructed. Image resolution can be taken into account by neglecting high-frequency coefficients expressing details smaller than pixel resolution, adjusting the rendered image's level of detail. Progressive decoding features coefficient reconstruction and refinement in order of greatest impact on rendering quality, allowing continuous adjustment of rendering quality vs. decoding time. With regard to rendering frame rate, decoding can be interrupted at any time. Aliasing artifacts are avoided as the inverse wavelet transform automatically interpolates inter-pixel intensity values.

Fig. 3 depicts progressively reconstructed light-field views. Image quality increases continuously with the number of bits decoded. While the view point is moving, images can be rendered at higher frame rate and somewhat lower resolution. During stand-still, high-quality images are rendered. Rendering quality can also be adjusted with regard to available computational power, yielding optimum results on fast hardware, but also providing interactive rendering rates on low-end computers.



0.006 bits per pixel (4000:1 compression)

0.03 bits per pixel (800:1 compression)

0.24 bits per pixel (100:1 compression)

Original Image

Figure 3: Progressive light-field decoding: image quality improves continuously during bitstream decoding. At 100:1 compression ratio, the reconstructed images are indistinguishable from the original images.

4 Coding Performance

The proposed light-field coding and rendering scheme is validated using two publicly available light fields[1]. The *Buddha* and *Dragon* test data sets consist of 32×32 24-bit RGB images with 256×256 pixels. The R, G and B channels are separately coded. Light-field reconstruction quality is measured using the Peak-Signal-to-Noise Ratio (PSNR) between the original and the reconstructed light-field by averaging over the 3 color channels of all images. Compression efficiency is expressed as the number of coding bits divided by the number of pixels in the light field $S \times T \times U \times V$ (bits per pixel).

Both light fields are coded at 0.24 bits per pixel, equivalent to 100:1 initial data compression. The bitstream is progressively decoded, and the reconstruction quality is measured for different bit-rates. Fig. 4 depicts coding performance for both light fields. For comparison, compression results of the disparity-compensating codec are shown [8]. Wavelet-based coding is not quite as efficient as disparity-compensated compression, especially at higher bit-rates. This is not surprising as the presented coder doesn't correct for parallax-induced differences between images. The 4D-Wavelet coder, however, does not introduce dependencies among light-field images which allows faster access to arbitrary light-field segments during decoding.

The *Buddha* light field is more efficiently compressed than the *Dragon* data set, indicating that coding efficiency is dependent on light-field scene content. The *Dragon* object shows more small-scale details than the *Buddha* statue, resulting in more high-frequency wavelet coefficients to be coded for high-quality reconstruction.

5 Discussion and Future Work

A combined coding and rendering scheme for light fields has been presented. Compression

---

[1] Available at:
www-graphics.standford.edu/software/lightpack/lifs.html

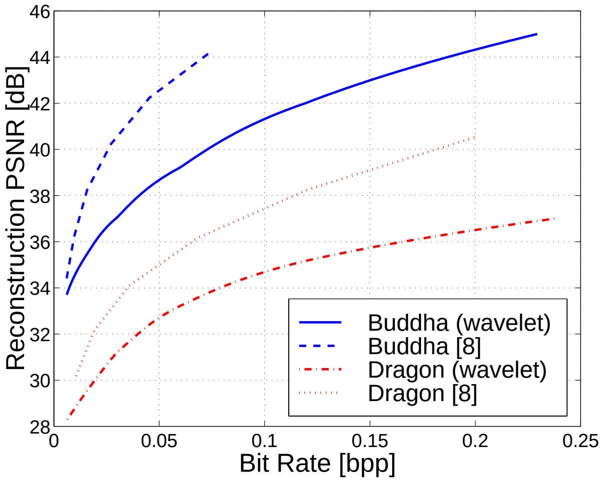


Figure 4: Coding performance for the light fields *Buddha* and *Dragon*. For comparison, compression results of the disparity-compensating codec [8] are depicted.

ratios exceeding 1000:1 are achieved at acceptable rendering quality. Progressive decoding allows continuous adjustment of rendering quality vs. frame rate.

The presented scheme can be easily extended to even higher-dimensional data structures. Time-varying light-field scenes or multiple light field recordings under changing illumination conditions, capturing bidirectional reflectance distributions (BRDF), can be efficiently compressed, transmitted, stored and rendered by a suitably adapted version of the proposed 4D-Wavelet coder.

Light fields will find their way into Internet commercials, computer games, and 3-D television once attractive LFR becomes available on standard home computers. The ability to handle large light-field data sets in conjunction with adjustable rendering quality paves the way towards light-field rendering on standard hardware equipment.

References

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

[2] S. Gortler, R. Grzeszczuk, R. Szelinski, and M. Cohen. The Lumigraph. *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 43–54, August 1996.

[3] A. Beers, M. Agrawala, and N. Chaddha. Rendering from compressed textures. *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 373–378, August 1996.

[4] G. Miller, S. Rubin, and D. Ponceleon. Lazy decompression of surface light fields for precomputed global illumination. *Rendering Techniques '98 (Eurographics Proceedings)*, pages 281–292, June 1998.

[5] T.-T. Wong, P.-A. Heng, S.-H. Or, and W.-Y. Ng. Image-based rendering with controllable illumination. *Eurographics Rendering Workshop 1997*, pages 13–22, June 1997.

[6] P. Lalonde and A. Fournier. Interactive rendering of wavelet projected light fields. *Proc. Graphics Interface'99*, pages 107–114, June 1999.

[7] I. Peter and W. Strasser. The wavelet stream - progressive transmission of compressed light field data. *Proc. IEEE Visualization'99*, October 1999.

[8] M. Magnor and B. Girod. Hierarchical coding of light fields with disparity maps. *Proc. International Conference on Image Processing (ICIP-99)*, Kobe, Japan, pages 334–338, October 1999.

[9] M. Magnor and B. Girod. Data compression for light field rendering. *IEEE Trans. Circuits and Systems for Video Technology*, April 2000.

[10] J. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Processing*, 41(12):3445–4362, December 1993.

[11] A. Said and W. Pearlman. A new, fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits and Systems for Video Technology*, 6(3):243–250, June 1996.