THEORETICAL ANALYSIS OF GEOMETRY INACCURACY FOR LIGHT FIELD COMPRESSION

Prashant Ramanathan and Bernd Girod

Information Systems Laboratory
Stanford University
{pramanat,bgirod}@Stanford.EDU

ABSTRACT

In current light field compression algorithms, geometry information is critical for efficient compression of light field image data. Inaccuracies in the geometry can greatly affect the compression efficiency. In this paper, we present a theoretical framework with which we can analyze and understand the effects of geometry inaccuracy on the compression efficiency of a light field coder. This framework combines ideas from statistical signal processing and rate-distortion theory. We demonstrate using simulations that the theory qualitatively matches the actual results of light field coding algorithms.

1. INTRODUCTION

Image-based rendering has emerged as an important new alternative to traditional image synthesis techniques in computer graphics. With image-based rendering, scenes can be rendered by sampling previously acquired image data, instead of synthesizing them from light and surface shading models and scene geometry. Light field rendering [1, 2] is one such image-based technique that is particularly useful for interactive applications.

A light field is a 4-D data set which can be parameterized as a 2-D array of 2-D light field images. For photo-realistic quality, a large number of high-resolution light field images is required, resulting in extremely large data sets. For example, the light field of Michelangelo's statue of *Night* contains tens of thousands of images and requires over 90 Gigabytes of storage for raw data [3]. Compression is therefore essential for light fields.

Several different light field compression techniques have been proposed. Vector quantization [1, 4] leads to faster rendering, whereas better compression efficiency is possible with block-based disparity-compensation [5], model-aided predictive coding [6], texture-based coding [7], or surface light fields [8]. The most efficient techniques use some form of geometry information, either by specifying depth values on a per-block basis, or using an explicit geometry model.

The accuracy of the geometry information can greatly affect the efficiency of the light field coder [9, 10]. The degree to which the compression efficiency is affected depends on the coding algorithm and how geometry is used. In this paper, we present a theoretical framework for analyzing the effect that geometry errors have on the compression efficiency of a particular light field coding algorithm. In particular, we assume a simplified geometry model, and a statistical model for geometry errors and image texture information. This analysis is inspired by the wide-sense

stationary theory proposed to analyze motion-compensation error in video compression [11].

The paper is organized as follows. Section 2 reviews light field compression algorithms. In Section 3, we present our theoretical model. In Section 4, we present simulation results for our theoretical model, and compare them with actual light field coding results. We conclude with suggestions for future work in Section 5.

2. GEOMETRY-BASED LIGHT FIELD COMPRESSION

Disparity compensation is used in several light field compression algorithms. The underlying idea of disparity-compensated prediction is that a pixel in a light field image can be predicted from corresponding pixels in one or more other light field images. This prediction requires a depth value for a given pixel. In a light field, the recording camera positions and orientations are known, which means that by specifying the depth, it is possible to establish correspondence between pixels in two different views. This pixel correspondence allows for the prediction of pixels of one view from another. We assume that the imaged surface looks similar in both views, which is true if the surface is Lambertian and is not occluded in either view.

In block-based disparity compensation, a depth value is specified for a block of pixels, typically selected from a predetermined set of depth values. In a model-aided prediction scheme, depth values are inferred from an explicit geometry model by rendering the model. In both cases, images are predicted from already encoded *reference* views. The order in which images are encoded, and the reference views that are used to predict a particular image are predetermined; the hierarchical coding structure [5], for example, can be used.

Texture-based coding [7] represents a different method of incorporating geometry information into the coding algorithm. When an explicit geometry model is available, each light field view can be mapped onto the geometry to obtain a view-dependent texture map. This set of texture maps, over the hemisphere of viewing directions, constitutes a 4-D data set that has greater correlation between views than the original 4-D data set.

Both the prediction-based scheme and the texture-based scheme are affected by geometry errors. For the prediction-based scheme, an incorrect depth value means that a pixel is predicted from the wrong pixel in the reference view; this may cost bits in the residual encoding. In the texture-based scheme, incorrect geometry means that pixels at the same position in the texture maps are not as correlated as they should be, which means more bits must be spent. In both cases, the degree to which this affects compression

efficiency depends on the magnitude of the geometry error and the local properties of the texture-image on the surface of the object. In the next section, we present a unified framework with which we can analyze these issues.

3. THEORETICAL MODEL

3.1. Signal And Encoding Model

We use a planar surface as a simplified model for geometry in our analysis. We assume that this surface has a 2-D texture signal $v(x, y)$, and is viewed by N cameras from directions $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N$. This arrangement is illustrated in Figure 1.

Fig. 1. The geometry is modelled as a plane in the x - y plane, with vertical error Δz . The back-projected image onto the model geometry results in the original texture shifted by (Δ_x, Δ_y) . This shift is dependent on the geometry error as well as the camera direction.

In addition, we assume that the surface is Lambertian. All non-Lambertian view-dependent effects are modelled as noise. The camera uses parallel projection and suffers no bandlimitation restriction due to imaging resolution limits. We can *backproject* the image in any of the views onto the geometry model to obtain exactly the texture that we started with.

As a result of these assumptions, the image signal is essentially the same whether in the texture or image plane. Due to this, prediction between images can be performed in the texture plane instead of the image plane. One advantage of this is that it allows us to compare different approaches, such as the texture-based scheme and the prediction-based scheme, more easily.

The geometry error is modelled as an offset Δz of the planar surface from its true position, as illustrated in Figure 1. When the image is backprojected from view i onto this inaccurate geometry, this results in a texture c_i that is a shifted version of the original texture signal v . We also include image noise and non-Lambertian view-dependent effects, both modelled as additive noise, to this shifted texture image giving us the equation

$$c_i(x, y) = v(x - \Delta_{xi}, y - \Delta_{yi}) + n_i(x, y) \quad (1)$$

where n_i is the additive noise component.

The shift, which depends only upon the camera's viewing direction $\mathbf{r}_i = [r_{ix} \ r_{iy} \ r_{iz}]^T$ and the geometry error Δz , is described by the equation

$$\begin{bmatrix} \Delta_x \\ \Delta_y \end{bmatrix} = \frac{\Delta z}{r_{iz}} \begin{bmatrix} r_{ix} \\ r_{iy} \end{bmatrix}. \quad (2)$$

Our eventual goal is a frequency-domain analysis of these signals, so we represent the shift with the transfer function

$$D_i(\omega_x, \omega_y) = e^{-j(\omega_x \Delta_{xi} + \omega_y \Delta_{yi})}. \quad (3)$$

The image vector $\mathbf{c} = [c_1 \ c_2 \ \dots \ c_N]^T$ represents the set of light field images or texture maps that have already been compensated or corrected with our (inaccurate) knowledge of the geometry. A light field coder does not encode these geometry-compensated images directly, but rather tries to exploit the correlation between them. Note that perfect knowledge of the geometry would mean that the geometry-compensated images match up perfectly.

Consider, for example, the prediction-based scheme. Here, we predict one image from another, and encode the residual error. In other words, we encode the first image and then encode the difference between the second image and the first. This is simply a linear transformation of the image vector $\mathbf{c}$ which gives us the error image vector $\mathbf{e} = [e_1 \ e_2 \ \dots \ e_N]^T$. We encode this set of images independently, instead of the original. The texture-based coder also uses a linear transformation to exploit correlation between the images $\{c_i\}$. Specifically, a Haar wavelet subband decomposition is used on $\mathbf{c}$ to obtain our error vector $\mathbf{e}$.

The block diagram in Figure 2 illustrates the various stages for a generic geometry-based light field coder. The texture v is observed in N different views. Each observation is is geometry-compensated to give the set of texture images $\{c_i\}$; each c_i includes additive image and view-dependent noise n_i . This set of images undergoes a linear transformation specified by the coding algorithm to give the set of error images $\{e_i\}$ which are independently encoded. In the next two subsections, we describe a statistical model for our image signals, and show how we can use this to evaluate the performance of various light field coding schemes.

Fig. 2. The signals c_i are produced by shifting the original texture signal v by $(\Delta_{xi}, \Delta_{yi})$ (described by the transfer function $D_i(\omega_x, \omega_y)$) and adding signal-independent noise n_i . The signals $\{c_i\}$ are then encoding by first linearly transforming them with matrix T to give the signals $\{e_i\}$. These signals are then independently encoded.

3.2. Statistical Model

We start by modelling our texture signal v as a wide-sense stationary random process with power spectral density (PSD)

$\Phi_{vv}(\omega_x, \omega_y)$. If we collect the set of shift transfer functions $\{D_i\}$ into a column vector

$$D = \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_N \end{bmatrix} = \begin{bmatrix} e^{-j(\omega_x \Delta_{x1} + \omega_y \Delta_{y1})} \\ e^{-j(\omega_x \Delta_{x2} + \omega_y \Delta_{y2})} \\ \vdots \\ e^{-j(\omega_x \Delta_{xN} + \omega_y \Delta_{yN})} \end{bmatrix} \quad (4)$$

then the power spectrum of vector signal $\mathbf{c}$ can be shown to be

$$\Phi_{cc} = \Phi_{vv} D D^H + \Phi_{nn} \quad (5)$$

where Φ_{nn} represents the power spectrum of the noise vector $\mathbf{n} = [n_1 \ n_2 \ \dots \ n_N]^T$ and the superscript H represents the complex-conjugate transpose of a matrix. Since $\mathbf{c}$ and $\mathbf{n}$ are vectors of size N , both Φ_{cc} and Φ_{nn} are matrices of size $N \times N$.

In (5), our geometry error Δz which influences the vector D , is a deterministic quantity. If we let Δz be a random variable with a probability density function (pdf) p , we obtain the revised equation

$$\Phi_{cc} = \Phi_{vv} E\{D D^H\} + \Phi_{nn} \quad (6)$$

where

$$E\{D D^H\} = \begin{bmatrix} 1 & \dots & P(\Omega^T(\mathbf{a}_1 - \mathbf{a}_N)) \\ P(\Omega^T(\mathbf{a}_2 - \mathbf{a}_1)) & \dots & P(\Omega^T(\mathbf{a}_2 - \mathbf{a}_N)) \\ \vdots & \ddots & \vdots \\ P(\Omega^T(\mathbf{a}_N - \mathbf{a}_1)) & \dots & 1 \end{bmatrix} \quad (7)$$

$P(\omega)$ represents the 1-D Fourier Transform of the pdf p of the random variable Δz , $\Omega = [\omega_x \ \omega_y]^T$ is a vector quantity, and $\mathbf{a}_i = \frac{1}{r_{iz}} [r_{ix} \ r_{iy}]^T$ depends on the original viewing direction vectors $\mathbf{r}_i$.

The power spectrum of the error image signal $\mathbf{e}$ is

$$\begin{aligned} \Phi_{ee} &= T \Phi_{cc} T^H \\ &= \Phi_{vv} T E\{D D^H\} T^H + T \Phi_{nn} T^H \end{aligned} \quad (8)$$

where T is the linear transformation matrix described in the previous section. Again, note that Φ_{ee} is a matrix of size $N \times N$.

3.3. Performance Measure

The power spectral density gives us the spatial correlation of a signal. As a consequence, we can use it to calculate the relative number of bits to encode one signal instead of another. If we assume that $\mathbf{c}$ and $\mathbf{e}$ are stationary, Gaussian signals with the same bandwidth, then we can calculate the bit-rate savings using the *rate-difference* equation [12]

$$\Delta R_i = \frac{1}{8\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \log_2 \left(\frac{\Phi_{eei}(\omega_x, \omega_y)}{\Phi_{cci}(\omega_x, \omega_y)} \right) d\omega_x d\omega_y. \quad (9)$$

ΔR_i represents the savings in bits per sample from encoding the signal e_i instead of c_i at high-rate. We average this expression over the entire set of images to obtain the overall rate-difference $\Delta R = \frac{1}{N} \sum_{i=1}^N R_i$.

We compare the performance of different light field coding algorithms, by comparing their overall rate-differences. This bit-rate savings depends on the PSD Φ_{ee} and, consequently, on the transformation matrix T that is specified by the algorithm.

4. RESULTS

Our theoretical framework reveals the effect of geometry inaccuracy on compression performance for different light field coding schemes. We compare two schemes: prediction-based and texture-based.

In [9, 10], experimental results from the two schemes on various different light field data sets are presented. The relevant observations are summarized here. The prediction-based scheme is more efficient at lower levels of geometry accuracy, whereas the texture-based scheme is more efficient for very accurate geometry. Thus, the texture-based scheme appears to be more sensitive to geometry errors than the prediction-based scheme. Our simulation results of the rate-difference of each scheme for various geometry accuracies show qualitatively the same trend.

To evaluate the rate-difference equation for various geometry accuracies, 256 views are arranged in a regular hierarchical structure over the hemisphere [5]. The random geometry error Δz is assumed to be a zero-mean Gaussian pdf with variance σ^2 .

The noise components that are added to each image are considered to be independent of one another. Since the results are sensitive to the overall level of noise, but not to the actual shape, we model the noise signal spectrum as having the same shape as the image signal spectrum. This is convenient since, as a result, the image signal PSD Φ_{vv} does not appear in the final expression. The noise and image spectrums are related by the equation $\Phi_{nni}(\omega_x, \omega_y) = \alpha \Phi_{vv}(\omega_x, \omega_y)$, where α is the ratio of noise signal variance to image signal variance. We arbitrarily use the value $\alpha = 0.001$ in our simulations.

The linear transformation matrix T , which describes the mapping from $\{c_i\}$ to $\{e_i\}$, is specified by the particular coding algorithm. For the texture-based scheme, the matrix entries correspond to the coefficients of the Haar wavelet subband decomposition. For the prediction-based scheme, the matrix entries are unity along the diagonal. If an image is encoded as is, and not predicted, then the corresponding row in the matrix will have all other non-diagonal entries as zero. If the image is predicted, then the corresponding row will contain negative entries at column locations corresponding to the reference images. The entries in a row for a predicted image will sum to zero.

Figure 3 plots the rate difference for the prediction and texture based schemes in bits per pixel versus geometry accuracy. Geometry accuracy is characterized by the equation $\log_2(\sqrt{12}\sigma)$, where σ is the square root of the variance of the geometry error. This expression, used in motion-compensation analysis for video compression [11], represents integer-pixel accuracy if the quantity is 0, half-pixel accuracy if the quantity is -1 , quarter-pixel accuracy if the quantity is -2 , and so on.

In the right-hand side of the plot, the low geometry accuracy region, we see that the prediction-based scheme outperforms the texture-based scheme, matching empirical observations in [9, 10]. Here, the misalignment between the texture images due to the shifts in the texture images is the dominant factor that affects the performance of the coding schemes. The prediction-based scheme predicts from views that have similar viewing direction, therefore the relative misalignment is smaller than in the texture-based scheme which encodes all views simultaneously. A similar geometric argument is used in [10] to explain the experimental results.

For very high geometry accuracy, there is very little misalignment due to geometry error, so, instead, the noise effect dominates the analysis. The texture-based coding scheme employs a

Fig. 3. Rate-difference [bits/pixel] vs. Geometry Accuracy [$\log_2(\sqrt{12}\sigma)$] for the prediction-based scheme and the texture-based scheme. For high geometry accuracy (left side of plot), the curve for the texture-based scheme is lower (more efficient) than the curve for the prediction-based scheme. Conversely, for less accurate geometry (right side of plot), the prediction-based scheme performs better. The ratio of noise variance to signal variance is $\alpha = 0.001$.

unitary transform that preserves the noise variance. In the case of the prediction-based scheme, forming the difference of two images increases the noise variance, and therefore degrades performance.

5. CONCLUSIONS

We have presented a theoretical framework to analyze the relationship between geometry inaccuracy and compression efficiency for light field compression. Using this framework, we compared the prediction-based and texture-based compression schemes and their sensitivities to geometry errors. Our theoretical simulation results qualitatively agree with actual coding experiments.

This work opens up many interesting questions and opportunities. One limitation of the framework presented is that we do not account for the resolution limits of the imaging device. By introducing bandlimitation to our framework, we can improve our model. Also, one hopeful goal of this research is to design more efficient light field coding algorithms. Finding the theoretically optimal transformation matrix may lead us in the right direction.

6. REFERENCES

- [1] Marc Levoy and Pat Hanrahan, "Light field rendering," in *Computer Graphics (Proceedings SIGGRAPH96)*, August 1996, pp. 31–42.
- [2] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen, "The lumigraph," in *Computer Graphics (Proceedings SIGGRAPH96)*, August 1996, pp. 43–54.
- [3] Marc Levoy, Kari Pulli, et al., "The Digital Michelangelo project: 3D scanning of large statues," in *Computer Graphics (Proceedings SIGGRAPH00)*, August 2000, pp. 131–144.
- [4] Xin Tong and Robert 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.
- [5] Marcus Magnor and Bernd Girod, "Data compression for light field rendering," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 3, pp. 338–343, April 2000.
- [6] Marcus Magnor, Peter Eisert, and Bernd 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.
- [7] Marcus Magnor and Bernd 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.
- [8] Daniel N. Wood, Daniel I. Azuma, et al., "Surface light fields for 3D photography," in *Computer Graphics (Proceedings SIGGRAPH00)*, August 2000, pp. 287–296.
- [9] Marcus Magnor, *Geometry-Adaptive Multi-View Coding Techniques for Image-based Rendering*, Ph.D. thesis, University Erlangen-Nuremberg, Germany, 2001.
- [10] Marcus Magnor and Bernd Girod, "Sensitivity of image-based and texture-based multi-view coding to model accuracy," in *Proceedings of the IEEE International Conference on Image Processing ICIP-2001*, Thessaloniki, Greece, October 2001, vol. 3, pp. 98–101.
- [11] Bernd Girod, "Efficiency analysis of multihypothesis motion-compensated prediction for video coding," *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 173–183, February 2000.
- [12] N.S. Jayant and Peter Noll, *Digital Coding of Waveforms*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.