IEEE CONFERENCE PROCEEDINGS: INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP2000)
VANCOUVER, CANADA, VOL. 2, PP. 5-8, SEPTEMBER 2000

TWO APPROACHES TO INCORPORATE APPROXIMATE GEOMETRY INTO MULTI-VIEW IMAGE CODING

Bernd Girod

Information Systems Laboratory
Stanford University
girod@ee.stanford.edu

Marcus Magnor

Telecommunications Laboratory
University of Erlangen-Nuremberg
magnor@LNT.de

INVITED PAPER

ABSTRACT

Two compression schemes for coding multiple images of a static scene are compared. Both codecs apply approximate 3-D scene geometry: in *Model-aided Coding*, images are predicted by model-driven disparity compensation, while the *Model-based Coder* uses geometry to convert images into texture maps for compression. Images of real-world objects are used to evaluate both codecs. Coding performance is examined subject to varying geometry accuracy. Compression factors of up to 1000:1 at acceptable reconstruction quality verify the usefulness of 3-D geometry for coding multi-viewpoint imagery.

1. INTRODUCTION

Today's computer rendering systems rely on 3-D scene descriptions, texture information and illumination specifications to calculate scene appearance from arbitrary viewpoints. Graphics hardware is available to accelerate *Geometry-based Rendering*. Despite computationally expensive calculations, however, rendering results often give a synthetic and unnatural impression.

For rendering photo-realistic views of real-world scenes, a novel approach has recently attracted considerable attention. *Image-based Rendering (IBR)* directly exploits information on scene appearance. IBR techniques use conventional images recorded from multiple viewpoints. IBR schemes range from solely image-based *Light Field Rendering (LFR)* [1] over *Lumigraph rendering* [2] exploiting local scene depth information to compensate disparity, to *View-Dependent Texture Mapping (VDTM)* [3], where different textures are applied to an approximate 3-D geometry model.

To attain photo-realistic rendering results, many hundreds to thousands of images are required for adequately sampling scene appearance. Data compression is therefore a vital issue in IBR. Efficient codecs for light fields and lumigraphs have been presented earlier [4], exploiting inter-image similarities by block-based disparity compensation. As no IBR-supporting graphics hardware exists up to date, LFR as well as Lumigraph rendering systems require high-end workstations to achieve interactive rendering rates, leaving little computational resources for image data decoding. If scene geometry is available, not only can disparity be accurately compensated, but occlusions are predictable as well, yielding better image estimates and enhanced compression. Scene geometry can additionally be used to exploit conventional 3-D graphics hardware.

In the following, two coding schemes are compared that use approximate 3-D geometry to compress multiple images of a static scene. Prior to coding, a volumetric geometry model of the scene is reconstructed using solely calibrated images. The high-resolution voxel model is approximated and converted to a triangle mesh for coding. The *Model-aided Coder* predicts images using 3-D geometry for disparity compensation and occlusion detection [5]. The residual prediction error is coded which allows reconstructing image regions outside the approximate model's silhouette. All images are coded in hierarchical order to efficiently exploit similarities among many images, introducing multiple dependencies during decoding. The *Model-based Coder* employs scene geometry to convert recorded images into texture maps [6]. Points on the object's surface correspond to fixed coordinates in the texture map, thus maximizing similarity between texture maps generated from different images. All texture maps are coded jointly using a subband coding scheme which

voxel model | 32 triangles | 128 triangles | 512 triangles | 2048 triangles

Figure 1: Reconstructed voxel model and approximate surface geometry models of the *Penguin*.

allows progressive reconstruction of arbitrary texture segments during decoding. Standard graphics hardware can be used to render the triangle mesh model with reconstructed texture.

2. GEOMETRY RECONSTRUCTION

To test coding performance, image sets of stuffed toy animals are recorded (Fig. 2). A turntable and a digital camera on a computer-driven lever arm are used to capture object appearance from multiple viewpoints. Internal and external camera parameters are calibrated by recording a calibration object from the same camera positions as the object images. To reconstruct 3-D geometry consistent with all images, a volumetric reconstruction algorithm is applied [7]. The algorithm discretizes the volume enclosing the scene into voxels and projects surface voxels into all visible images. For each voxel, color hypotheses are collected and checked for consistency. A voxel is removed if none of its color hypotheses is in accordance with the image projections. Iteration over all remaining voxels ends when no further voxel can be removed (Fig. 1).

Triangulating the reconstructed model surface yields several hundred thousand triangles. Approximate geometry models are therefore generated by starting from a closed mesh consisting of only a few triangles: the initial mesh is placed at the center of the voxel model, and each vertex is moved along its normal direction until it hits the model surface. The mesh is refined by inserting new vertices at edge midpoints, subdividing each triangle into four new triangles (Fig. 3). Vertex normals are calculated from adjacent triangles, and the new vertices are again moved to the voxel model surface, yielding a better model approximation. By repeated triangle subdivision and vertex position refinement, increasingly accurate geometry models are obtained (Fig. 1). Triangle subdivision changes mesh topology in a deterministic way which allows efficiently projecting 3-D geometry to the planar texture map. To code the triangle mesh describing approximate object geometry, the *Embedded Mesh Coding (EMC)* algorithm is applied [8].

Figure 2: For evaluation, 3 objects are recorded from 257 different directions.

3. MODEL-AIDED CODER

The *Model-aided Codec* is described in more detail in [5]. Images are coded in hierarchical order, i.e., already coded images serve as reference to predict not-yet coded images, exploiting similarities between images. Geometry is used to predict scene appearance from nearby recorded images, compensating disparity and detecting occluded image regions. While only pixels within the projected model silhouette can be predicted, the prediction error is coded for the entire image which allows compensating for coarse model silhouette. But since the *Model-based Coder* can only reconstruct image pixels inside model silhouette, for comparison we have restricted the *Model-aided Coder* to code also only pixels within the projected silhouette. Consequently,

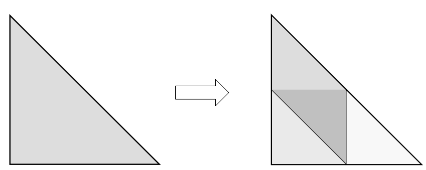

Figure 3: Approximate geometry is refined by inserting new vertices at edge midpoints, subdividing each face into four new triangles.

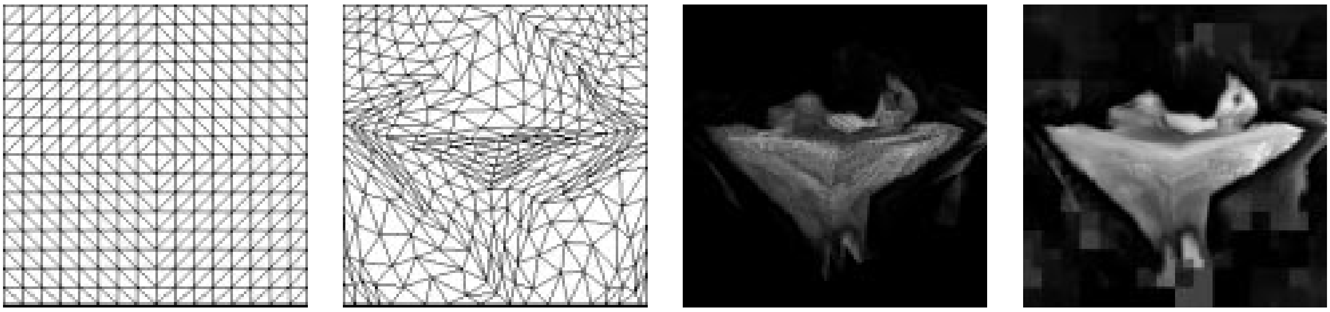

Figure 4: Texture map generation: the texture plane is regularly partitioned by triangle subdivision; vertex postions are moved to optimize triangle size; a sparse texture map is obtained by mapping the image onto the texture plane; missing texture information is interpolated.

all bit-rates and image quality measurements in the following are expressed with regard to the number of reconstructed object pixels.

4. MODEL-BASED CODER

To use graphics hardware for rendering acceleration, images must be converted into texture maps. Texture maps also exhibit beneficial coding characteristics: because texture-map coordinates correspond to object surface points, texture maps generated from different images show similar color values at the same texture element (texel) position. The *Model-based Coder* converts all images into texture maps for coding [6].

The approximate model's triangle mesh can be efficiently mapped onto the texture plane (Fig. 4). To adapt texture-map triangle size to actual geometry size, an optimization scheme is applied: vertex positions are moved until all texture triangles have about the same relative size as their corresponding geometry triangles. By mapping the images onto the texture plane, sparsely filled texture maps are obtained. Holes in the texture maps are interpolated from nearby texels. The texture maps are subband-decomposed, exploiting correlation within as well as between texture maps. A modified *SPIHT* algorithm [9] is used to progressively code the resulting 4-D wavelet coefficients [10].

5. MEASUREMENTS

Both coders are evaluated using the *Penguin*, *Mouse* and *Garfield*^{TM} image sets, each consisting of 257 384×288-pixel RGB images. Recording positions cover the upper half-sphere around the objects. In the following, bit-rates include geometry coding bits and are measured in *bits per* reconstructed object *pixel (bpp)*. All images are converted to YUV color space, and chrominance components are downsampled by a factor 2 in horizontal and vertical direction. Reconstruction quality is expressed as the luminance component's *Peak Signal to Noise Ratio (PSNR)*, averaged over all decoded images.

Fig. 5 depicts rate-distortion performance for different geometry accuracies of the *Penguin* object. For both coders, the optimal object model consists of only 512 triangles and is independent of target bit-rate. While more accurate geometry yields only marginally better image predictions, it increases the number of object pixels to code as well as the geometry coding bit-rate. Fig. 6 illustrates coding performance for all 3 image sets using geometry models of respectively optimal accuracy. At high bit-rates, the *Model-aided Coder* performs up to ~ 2 dB better than the *Model-based Codec*. To code the texture maps, MBC applies the Haar wavelet which only insufficiently exploits image signal characteristics. Reconstruction accuracy of MBC is also limited by texture-map size as image pixels

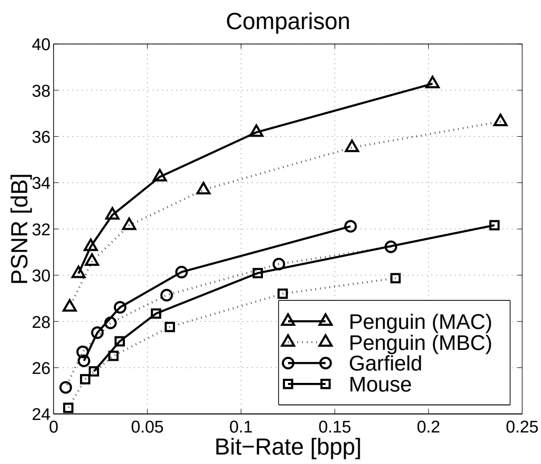

Figure 6: Performance of the *Model-aided Coder (MAC)* and *Model-based Coder (MBC)* for the *Penguin*, *Mouse* and *Garfield*^{TM} image sets using optimal geometry approximations.

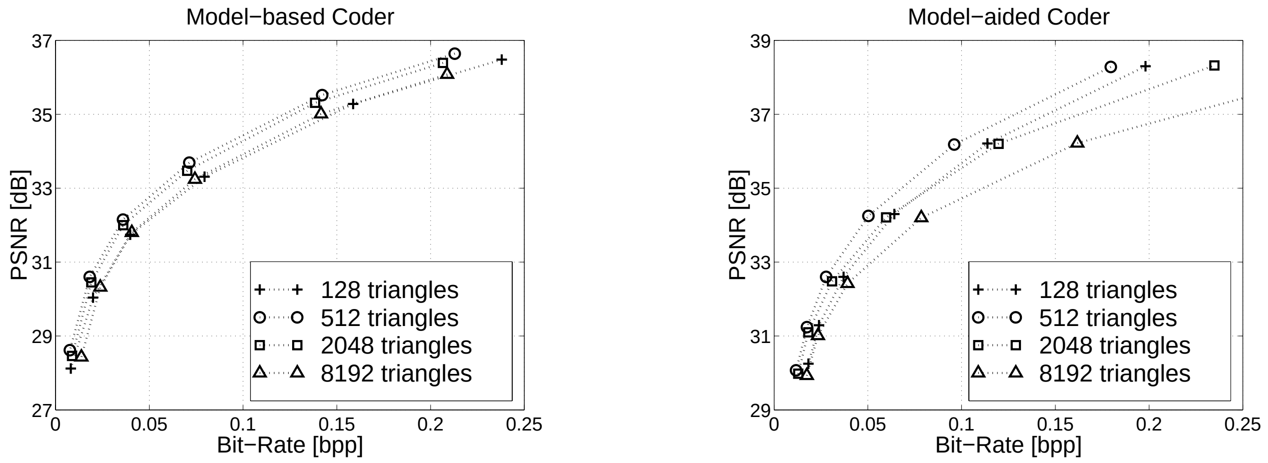

Figure 5: Rate-distortion curves of the *Model-aided* and *Model-based* codecs for the *Penguin* image set using different geometry approximations.

can be mapped onto the same texel.

6. CONCLUSIONS

Two codecs have been compared that apply 3-D geometry to code multiple images of a static scene. The *Model-aided Coder* hierarchically predicts images and codes residual prediction error. The *Model-based Coder* converts images into texture maps and applies subband coding for joint compression of texture information. The former coder achieves higher compression rates, while the latter features progressive decoding and direct access to arbitrary texture segments. Graphics hardware can be directly used in conjunction with model-based coding to accelerate rendering. The attained compression factors confirm the usefulness of approximate geometry for multi-view image coding.

7. REFERENCES

[1] M. Levoy and P. Hanrahan, “Light field rendering”, *Computer Graphics (SIGGRAPH ’96 Proceedings)*, pp. 31–42, Aug. 1996.

[2] S. Gortler, R. Grzeszczuk, R. Szelinski, and M. Cohen, “The Lumigraph”, *Computer Graphics (SIGGRAPH ’96 Proceedings)*, pp. 43–54, Aug. 1996.

[3] P. Debevec, C. Taylor, and J. Malik, “Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach”, *Computer Graphics (SIGGRAPH ’96 Proceedings)*, pp. 11–20, Aug. 1996.

[4] M. Magnor and B. Girod, “Data compression for light field rendering”, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 10, no. 3, pp. 338 – 343, Apr. 2000.

[5] M. Magnor, P. Eisert, and B. Girod, “Model-aided coding of multi-viewpoint image data”, *Proc. International Conference on Image Processing (ICIP-2000),* Vancouver, Canada, Sept. 2000.

[6] M. Magnor and B. Girod, “Model-based coding of multi-viewpoint imagery”, *Proc. Visual Communications and Image Processing 2000 (VCIP’2000),* Perth, Australia, June 2000.

[7] P. Eisert, E. Steinbach, and B. Girod, “Multi-hypothesis volumetric reconstruction of 3-D objects from multiple calibrated camera views”, *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP’99)* Phoenix, USA, pp. 3509–3512, Mar. 1999.

[8] M. Magnor and B. Girod, “Fully embedded coding of triangle meshes”, *Proc. Vision, Modeling, and Visualization (VMV’99),* Erlangen, Germany, pp. 253–259, Nov. 1999.

[9] 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*, vol. 6, no. 3, pp. 243–250, June 1996.

[10] M. Magnor and B. Girod, “Progressive compression and rendering of light fields”, *Proc. Vision, Modeling, and Visualization (VMV2000),* Saarbrücken, Germany, Nov. 2000, submitted.