University of Victoria

An Evaluation of Several Mesh-Generation Methods Using a Simple Mesh-Based Image Coder

Michael D. Adams

Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, V8W 3P6, Canada

mdadams@ece.uvic.ca

1. Introduction

The image representations used by most conventional image coders employ regular (e.g., lattice) sampling. Due to the nonstationary nature of most images, however, such sampling is usually highly inefficient. Inevitably, when regular sampling is employed, the sampling density will be too low in regions where the signal is rapidly changing, and too high in regions where the signal is varying slowly or not at all. The above problem can be overcome by employing an image representation that facilitates the use of arbitrary (i.e., irregular) sampling. In this context, image representations based on triangle meshes are of particular interest, as such meshes are ideally suited to accommodating arbitrary sampling. One of the major challenges associated with mesh-based representations is finding effective methods for constructing a mesh that accurately represents a given image (i.e., the so-called mesh-generation problem).

2. Objective

The goal of this work is to develop a simple mesh-based image coder and then use it to evaluate the performance of several mesh-generation methods. Through the results of this evaluation, we can gain a better understanding of the relative effectiveness of these methods as well as obtain some of the insight necessary to develop improved mesh-generation methods in the future.

3. Mesh-Based Image Representation

A grayscale image can be viewed as a surface defined on a planar domain. This image surface can, in turn, be approximated using a triangle mesh. In effect, the image domain is partitioned using a triangulation, and then an interpolant for the image surface is defined over each face of the triangulation. An image can be reconstructed from the triangle mesh via rasterization. The above image representation scheme is illustrated in Fig. 1.

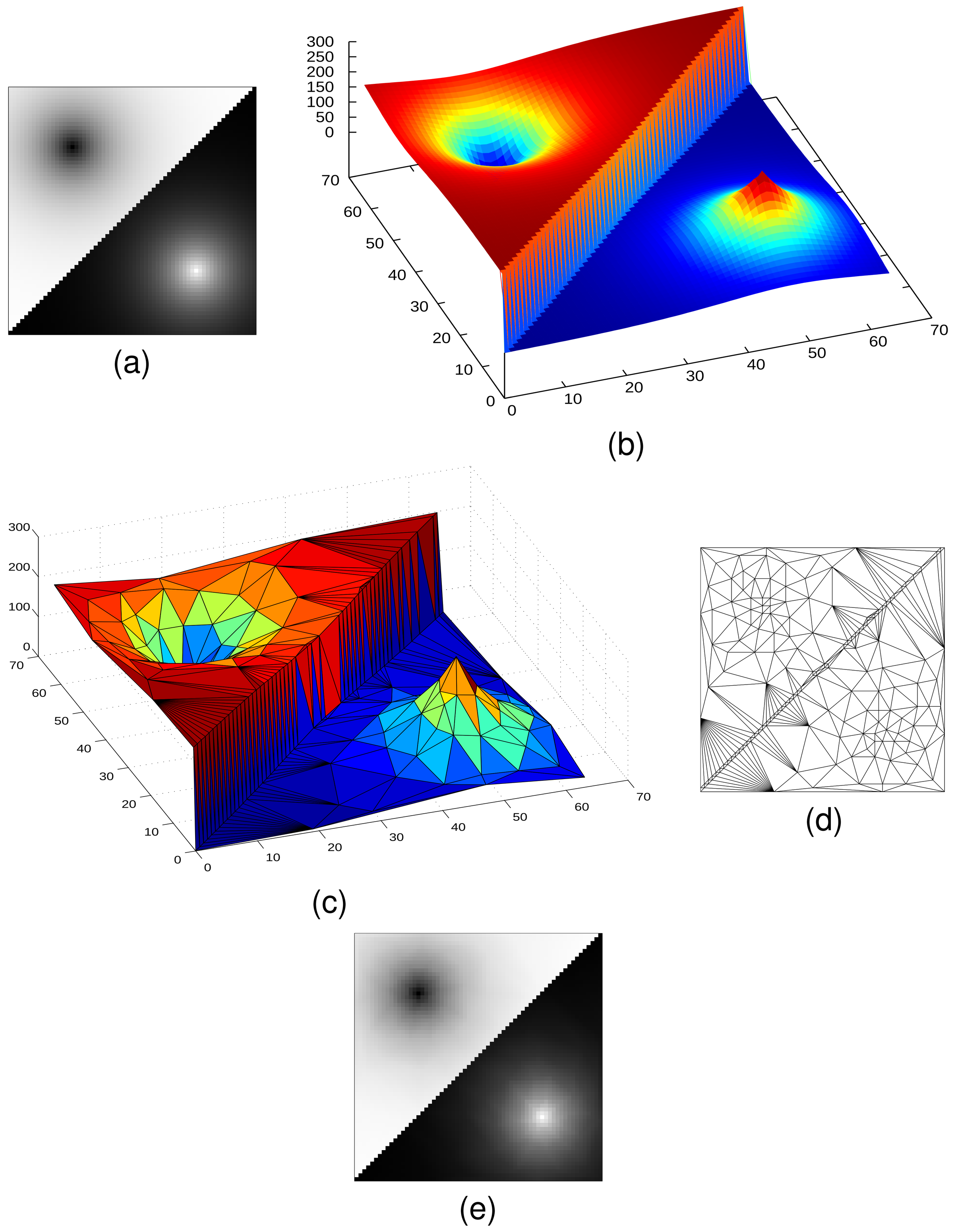

Fig. 1: *Mesh-based image representation example. (a) The original image and (b) its corresponding surface; (c) a triangle-mesh approximation of the image surface, (d) its corresponding image-domain triangulation, and (e) the image reconstructed from the triangle mesh.*

4. Mesh-Based Image Coder

To provide a framework for evaluating the performance of several mesh-generation methods, we have developed a simple mesh-based image coder. This coder supports both lossy and lossless compression of (grayscale) images of arbitrary width and height, and is based on the scattered data coding (SDC) method [1]. The general structure of the coder is shown in Fig. 2. The encoder, shown in Fig. 2(a), first constructs a mesh that well approximates the original image. Then, the resulting mesh is coded in some efficient manner. The decoder, shown in Fig. 2(b), first decodes the coded version of the mesh, and then converts the resulting mesh into a raster image. The rasterization scheme employed typically depends on the mesh-generation method used in the encoder. In our coder, the rate is controlled by adjusting the mesh vertex count (rather than through quantization of the mesh data).

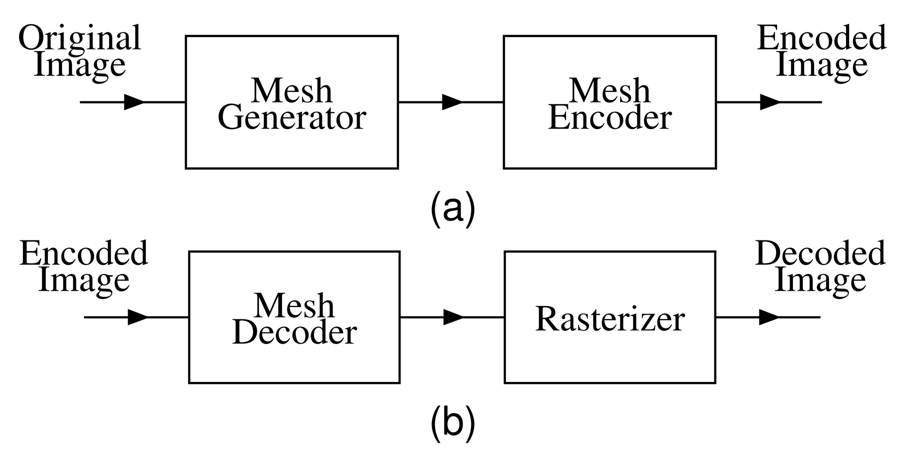

Fig. 2: *General structure of the mesh-based image coder. (a) Encoder and (b) decoder.*

5. Mesh-Generation Methods

In our work, we considered the following four mesh-generation methods:

1) Yang, Wernick, and Brankov (YWB) method [2]. In this method, a feature map is computed that approximates the largest magnitude second directional derivative at each point in the image. Then, Floyd-Steinberg error diffusion is used to distribute sample points so that their local spatial density is approximately proportional to the corresponding value in the feature map. The number of sample points is controlled indirectly through an error-diffusion threshold parameter. The sample points are triangulated in order to establish the mesh connectivity.

2) Garland and Heckbert (GH) method (i.e., data-independent greedy insertion from [3]). In this method, the four corners of the image bounding box are triangulated to form an initial approximation. Then, the (unused) sample point with the largest absolute error is inserted into the triangulation. This process is repeated until the vertex-count budget is exhausted.

3) Modified GH (MGH) method (i.e., our proposed method). In this method, the four corners of the image bounding box are triangulated to form an initial approximation. Then, we iterate as follows. In the current triangulation, select the triangle with the largest squared error. Within this triangle, choose the (unused) sample point with the largest absolute error and insert it into the triangulation. Repeat until the vertex-count budget is exhausted.

4) Random method. This very trivial method (used for benchmarking purposes) simply chooses sample points randomly and then triangulates them to form a mesh.

In all of the above mesh-generation methods, a linear interpolant is used to form a surface through the mesh vertices. We also consider variants of the GH and MGH methods that employ the Clough-Tocher (CT) interpolant [4], yielding what we refer to as the GH-CT and MGH-CT methods, respectively.

6. Results

Lossy coding results comparing the performance of the YWB, GH, MGH, and random mesh-generation methods are shown in Figs. 3, 4, and 5. From these results, we can see that our MGH method performs best, followed by the GH method, and then the YWB and random schemes. Note that the YWB scheme is quite ineffective at very low rates, performing poorer than the random method.

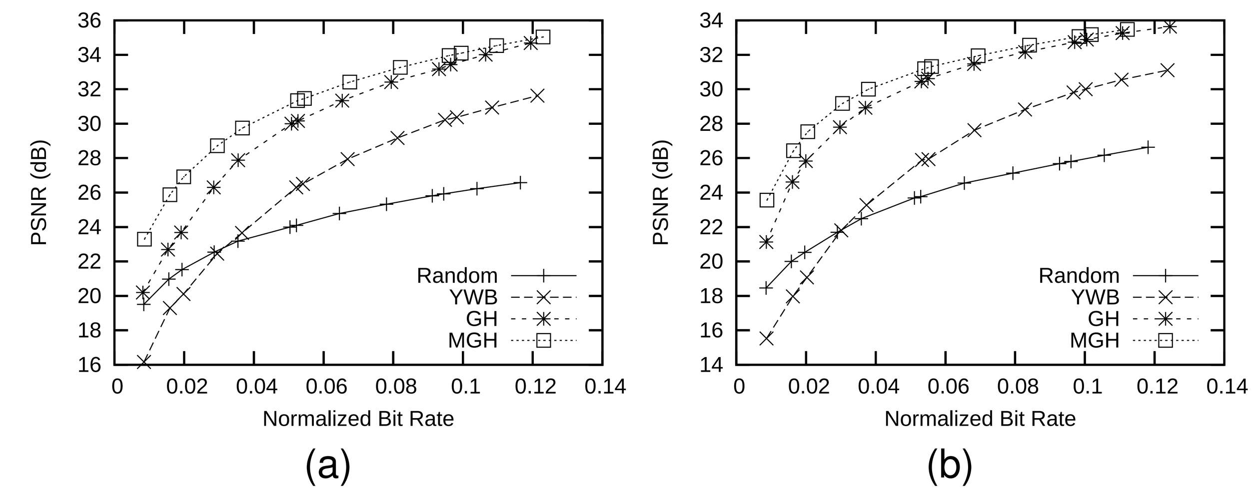

Fig. 3: *Lossy coding results for the (a)* `lena` *and (b)* `peppers` *images using the random, YWB, GH, and MGH methods.*

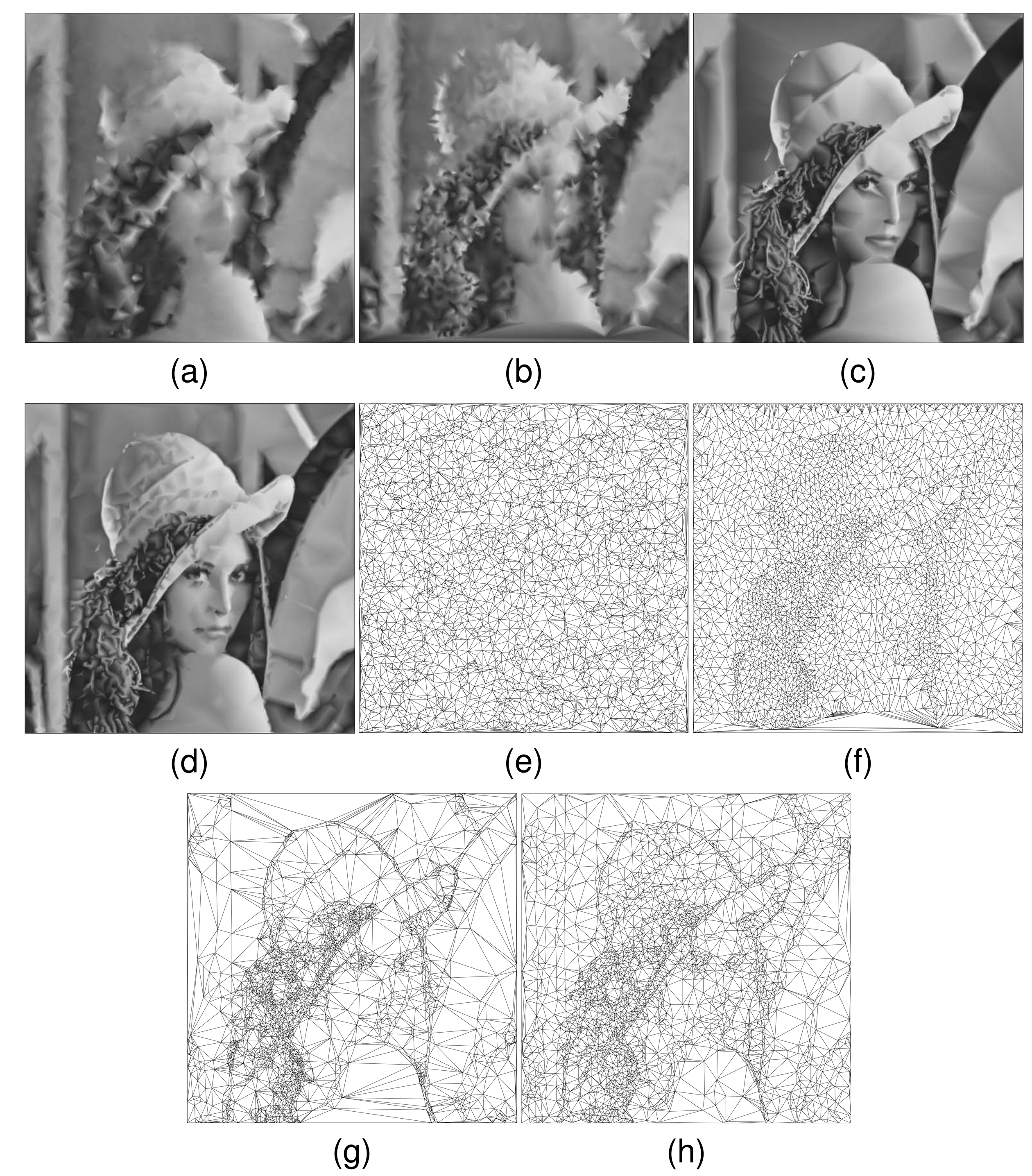

Fig. 4: *Lossy reconstructed images obtained at about 50:1 compression with the (a) random (21.52 dB), (b) YWB (20.10 dB), (c) GH (23.67 dB), and (d) MGH (26.91 dB) methods; and the image-domain triangulation associated with each of the (e) random, (f) YWB, (g) GH, and (h) MGH methods.*

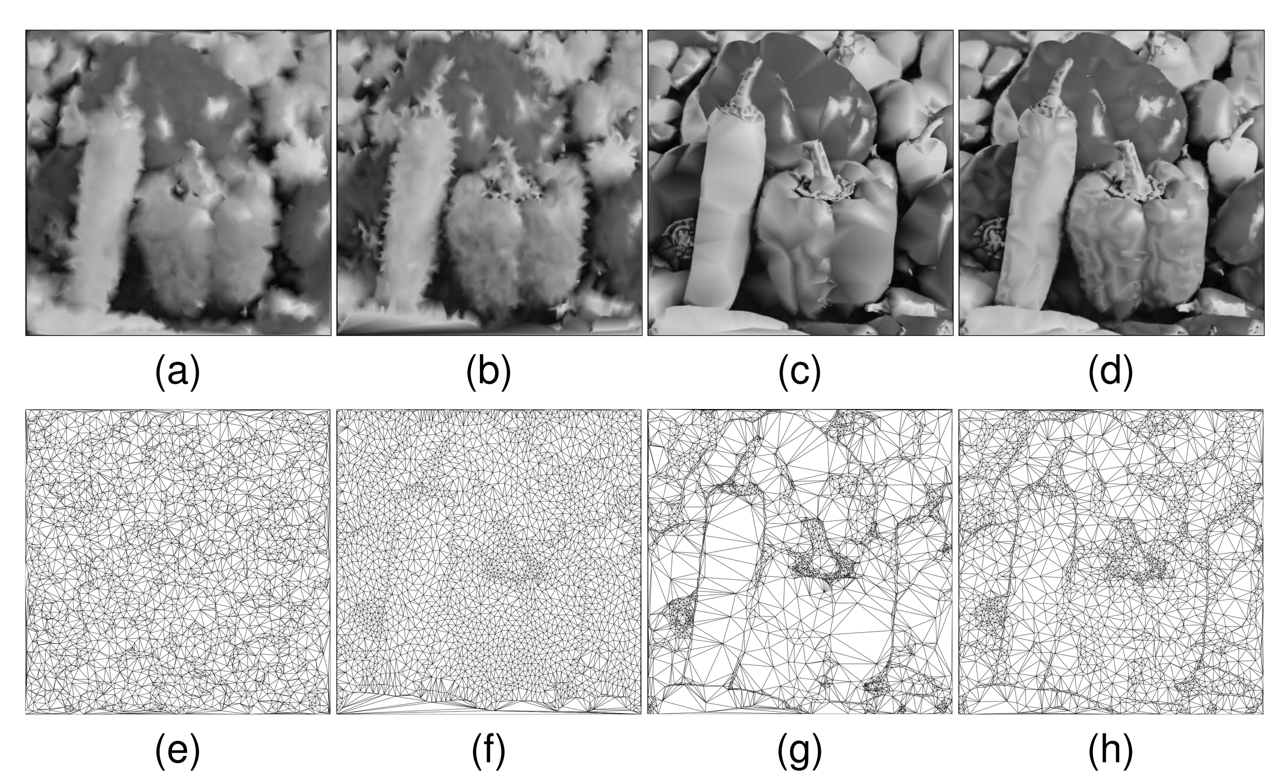

Fig. 5: *Lossy reconstructed images obtained at about 50:1 compression with the (a) random (20.52 dB), (b) YWB (19.07 dB), (c) GH (25.83 dB), and (d) MGH (27.53 dB) methods; and the image-domain triangulation associated with each of the (e) random, (f) YWB, (g) GH, and (h) MGH methods.*

Lossy coding results comparing the effectiveness of linear and CT interpolation are shown in Figs. 6 and 7. In particular, we compare the GH and MGH methods, which employ linear interpolation, to their variants employing CT interpolation, namely, the GH-CT and MGH-CT methods, respectively. From these results, we can see that the GH method outperforms the GH-CT method, and the MGH method outperforms the MGH-CT method. In other words, linear interpolation outperforms CT interpolation. The poor performance of the CT interpolant is largely due to severe overshoot/undershoot in the vicinity of image edges.

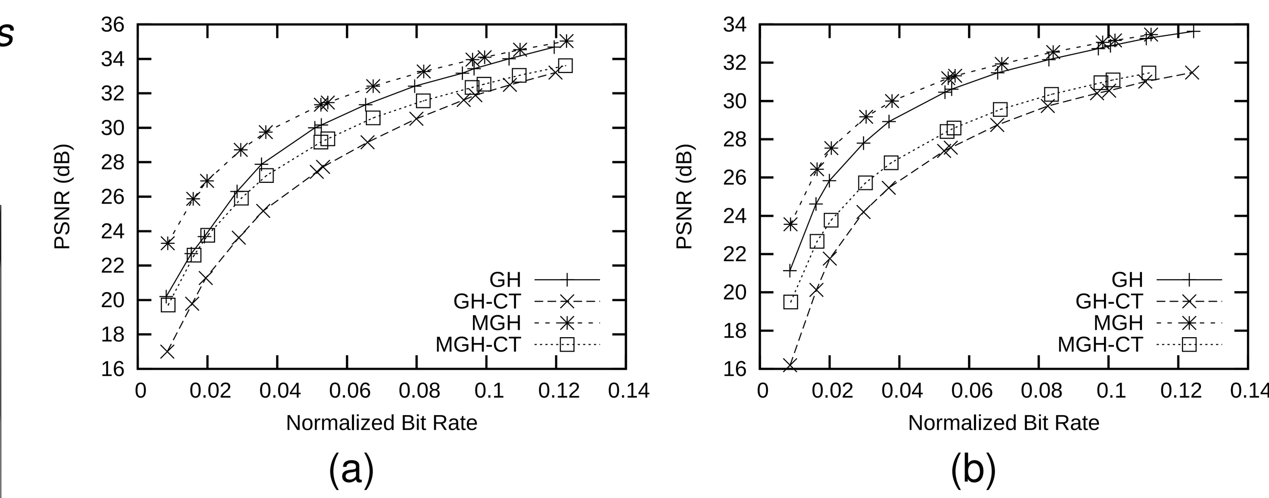

Fig. 6: *Lossy coding results for the (a)* `lena` *and (b)* `peppers` *images using the GH, GH-CT, MGH, and MGH-CT methods.*

Fig. 7: *Lossy reconstructed images obtained at about 50:1 compression with the (a) MGH (26.91 dB) and (b) MGH-CT (23.76 dB) methods.*

7. Conclusions

We have developed a simple mesh-based image coder and used this coder to evaluate the performance of several mesh-generation methods. Of the methods considered, our proposed MGH scheme was shown to perform best (both objectively and subjectively) by a significant margin. Through our evaluation, we have also shown that the use of a CT interpolant leads to much poorer results than a linear interpolant, due to severe overshoot/undershoot in the vicinity of image edges. Through the insights provided by our work, one can hope to develop improved mesh-based image coders in the future.

References

[1] L. Demaret and A. Iske, "Scattered data coding in digital image compression," in *Curve and Surface Fitting: Saint-Malo 2002*, Brentwood, TN, USA, 2003, pp. 107–117, Nashboro Press.

[2] Y. Yang, M. N. Wernick, and J. G. Brankov, "A fast approach for accurate content-adaptive mesh generation," *IEEE Trans. on Image Processing*, vol. 12, no. 8, pp. 866–881, Aug. 2003.

[3] M. Garland and P. S. Heckbert, "Fast polygonal approximation of terrains and height fields," Tech. Rep. CMU-CS-95-181, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, Sept. 1995.

[4] R. Clough and J. Tocher, "Finite element stiffness matrices for analysis of plates in bending," in *Proc. of Conference on Matrix Methods in Structural Mechanics*, Wright-Patterson AFB, OH, USA, 1965.