ICER on Mars

Wavelet-Based Image Compression for the Mars Exploration Rovers

Aaron Kiely, Matthew Klimesh, and Justin Maki

Introduction

In early 2004, the Mars Exploration Rover (MER) mission will land a pair of rovers on Mars and operate each of them for 90 Martian days. Each rover is equipped with an unprecedented nine visible-wavelength cameras: a mast-mounted, high-angular-resolution color imager for science investigations (the panoramic camera, or Pancam); a mast-mounted, medium-angular-resolution camera for navigation purposes (the Navcam); a set of body-mounted, front-and-rear cameras for navigation hazard avoidance (the Hazcams); and a high-spatial-resolution camera (the Microscopic Imager) mounted on the end of a robotic arm for science investigations. With the exception of the Microscopic Imager, all of these cameras are actually stereo camera pairs.

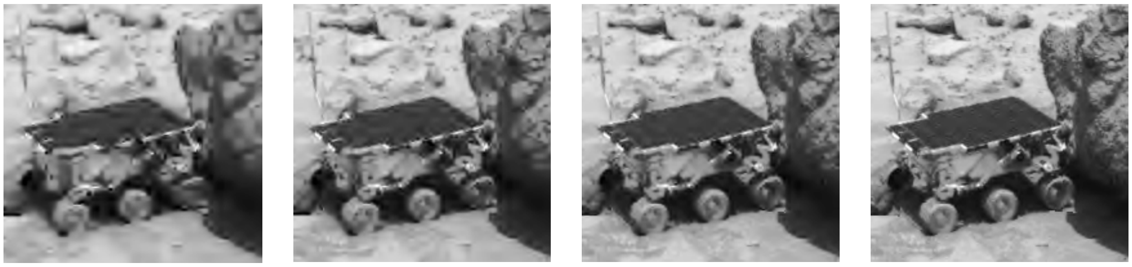
Not surprisingly, collecting and transmitting images to Earth will be a major priority of the mission: well over half of the bits transmitted from the rovers will consist of compressed image data from these cameras. For image compression, MER will rely exclusively on the ICER image compressor, developed in the Science Processing and Information Management work area at the Jet Propulsion Laboratory (JPL).

The development of ICER was driven by the desire to achieve state-of-the-art compression performance with software that meets the specialized needs of deep space applications. ICER's compression effectiveness will enhance the ability of the MER mission to meet its science objectives. In this article, we give a brief overview of the features and inner workings of ICER.

ICER Overview

ICER is a *wavelet-based* image compressor that features *progressive compression*. The ICER software can provide both lossless and lossy compression and incorporates a sophisticated error-containment scheme to limit the effects of data loss.

Progressive compression means that as more compressed data are received, successively higher quality reconstructed images can be reproduced, as illustrated in Figure 1. Thus, one could, for example, send a small fraction of the data from an ICER-compressed image to get a low-quality preview and later send more of the data to get a higher quality version if the image is deemed interesting. In future missions, progressive compression will enable sophisticated data return strategies involving incremental image quality improvements to maximize the returned science value using an on-board buffer.



0.125 bit/pixel

0.25 bit/pixel

0.5 bit/pixel

1.0 bit/pixel

Figure 1. This sequence of image details from a larger image shows how image quality improves under progressive compression as more compressed bits are received

A wavelet transform is a linear (or nearly linear) transform designed to decorrelate images by local separation of spatial frequencies. In wavelet-based image compression, a wavelet transform is applied to the image before further processing. By using a wavelet transform, ICER avoids the "blocking" artifacts that can occur when the discrete cosine transform (DCT) is used for decorrelation, as in the Joint Photographic Expert Group (JPEG) compressor used on the Mars Pathfinder mission. Also, wavelet-based compression is usually superior to DCT-based compression in terms of quantitative measures of reconstructed image quality.

When used for lossless image compression, ICER produces compressed image sizes that are competitive with those produced by state-of-the-art algorithms designed exclusively for lossless image compression (e.g., [7]) and about 20% smaller than the Rice compressor used for lossless image compression on Mars Pathfinder.

In a nutshell, here's how ICER works. A wavelet transform decomposes the image into several subbands, each a smaller version of the image, but filtered to contain a limited range of spatial frequencies. ICER achieves progressive compression by successively encoding bit planes within the subbands. (A bit plane consists of all of the bits of a given significance from the subband; for example, the second bit plane of a subband contains the second most significant bit from each member of the subband.) While encoding, ICER maintains a statistical model of the image that allows effective compression of the bit planes. Lossless compression is achieved when all of the compressed subband bit planes are transmitted.

For error containment purposes, the compressed data are organized into a user-selectable number of segments, each containing information about a limited rectangular region of the image. Each segment is compressed independently so that if one of the packets for a segment is lost due to channel noise, the other segments are unaffected.

Controlling Image Quality and Amount of Compression

All MER cameras produce 1024-pixel by 1024-pixel images at 12 bits per pixel. Images transmitted from MER will range from tiny 64 × 64 "thumbnail" images up to full-size images. MER will make use of compressed bit rates ranging from less than a bit/pixel up to rates yielding lossless compression. Current MER baseline plans call for navigation, thumbnail, and many other image types to be compressed to approximately 1 bit/pixel; lower bit rates may be used for certain wavelengths of multi-color panoramic images. At the other end of the compression spectrum, radiometric calibration targets are likely to be compressed to about 4 to 6 bits/pixel, and lossless compression will be used for certain science images.

To control the image quality and amount of compression in ICER, the user specifies a *byte quota* (the nominal number of bytes to be used to store the compressed image) and a *quality level* parameter (which is essentially a quality goal). ICER attempts to produce a compressed image that meets the quality level using as few compressed bytes as possible. ICER stops producing compressed bytes once the quality level or byte quota is met, whichever comes first.

This arrangement provides added flexibility compared to compressors (like the JPEG compressor used on Mars Pathfinder) that provide only a single parameter to control image quality. Using ICER, when the primary concern is the bandwidth available to transmit the compressed image, one can set the quality goal to lossless and the given byte quota will determine the amount of compression obtained. At the other extreme, when the only important consideration is a minimum acceptable image quality, one can provide a sufficiently large byte quota and the amount of compression will be determined by the quality level specified.

Wavelet Transform

The first step in ICER compression is to perform a two-dimensional wavelet transform of the image. This transform decomposes the image into a user-controlled number of subbands. To conserve memory on board MER, the wavelet-transformed image is stored using the same memory array as the original image.

The wavelet transform is used to decorrelate the image, concentrating most of the important information into a small number of small subbands. Thus, a good approximation to the original image can be obtained from a small amount of data. In addition, the subbands containing little information tend to compress easily. The wavelet transform does leave some correlation in the subbands, so compression of the subband bit planes uses predictive compression to attempt to exploit as much of this remaining correlation as possible.

The two-dimensional wavelet transform is accomplished through several applications of a one-dimensional wavelet transform to rows and columns of data. An ICER user can select one of several reversible integer wavelet transforms [1, 5]. Each is a nonlinear approximation to a linear high-pass/low-pass filter pair. The wavelet transform produces integer outputs and is exactly invertible, which allows us to achieve lossless compression when all of the subband data are reproduced exactly.

After the wavelet transform, ICER successively compresses bit planes of the subbands. Ideally, at each stage of compression, we would like to encode the subband bitplane giving the biggest improvement in some measure of image quality per compressed bit. However, performing this optimization on-board would be impractical and would achieve rather modest gains in compression effectiveness; therefore, in practice, ICER uses heuristics to select which subband bit plane to transmit at each stage.

Context Modeling

ICER employs a technique known as *context modeling* in its encoding of the bit planes. With this technique, before encoding a bit in the transformed image, the bit is classified into one of several *contexts* based on the values of previously encoded bits. These bits are from the pixel being encoded and nearby pixels and include the bits previously encoded from the current bit plane as well as bits from previous (more significant) bit planes. The probability that the bit to be encoded is a "0" is estimated based on the encoder's previous experience with bits classified into the same context. The bit is then encoded based on this probability estimate. Since the probability estimate relies only on previously encoded information, the decoder can duplicate this calculation and produce the same probability estimate, which is essential for proper decoding.

Although the above procedure may sound complicated, in fact it is done with reasonably low complexity. In particular, ICER's scheme for classification into contexts is fairly simple: it uses two small lookup tables, is based on simple properties of the pixel being encoded as well as the 8 nearest neighbors pixels, and yields one of 18 contexts. This scheme is similar to the context modeling used by JPEG2000 [6].

Entropy Coding

The actual compression of the bits is accomplished with an *entropy coder*. An entropy coder is a module that takes a sequence of bits along with corresponding probability-of-zero estimates for the bits, and produces an encoded (and hopefully compressed) bit stream from which the original sequence of bits can be recovered. At the decoding end, it's necessary for the decompressor to reconstruct the probability-of-zero estimate for each bit before decoding it.

Although binary entropy coding is usually performed with a technique known as *arithmetic coding* (or with a low-complexity approximation to arithmetic coding), we have chosen to use a lesser known technique called *interleaved entropy coding* [2, 3, 4]. We have produced a software implementation of interleaved entropy coding that has particularly low complexity; therefore, it is well-suited for space applications and other applications where speed can be of critical importance. Given perfect probability-of-zero estimates, both arithmetic coding and interleaved entropy coding can compress stochastic bit sequences to within 1% of optimal.

Although the entropy coding stage of ICER is extremely effective, improvements to the overall compression effectiveness should still be possible. The wavelet transform and the context modeling are designed to exploit as much correlation as possible from the image; however, this process is not an exact science.

Error Containment

Error containment is essential to accommodate bit error rates seen on the deep space channel. Without error containment, a single packet loss due to channel errors can corrupt large segments of compressed data.

To achieve error containment, ICER produces the compressed bitstream in separate pieces or *segments* that can be decoded independently. These segments represent rectangular regions of the original image, but are defined in the transform domain. If the image were partitioned directly and the wavelet transform separately applied to each segment, under lossy compression the boundaries between segments would tend to be noticeable in the reconstructed image even when no compressed data is lost, as illustrated in Figure 2(a). By segmenting the image in the transform domain, we can virtually guarantee that such artifacts will not occur, as illustrated in Figure 2(b). There are also secondary benefits: we achieve better decorrelation by applying the wavelet transform to the entire image at once and it is easier to maintain a similar image quality in the different segments. A minor side effect is that the effect of data loss in one segment can appear to “bleed” slightly into adjacent segments. Note that segments that are easier to compress will naturally use fewer bits, so lengths of compressed segments will be variable.

ICER includes an algorithm to automatically partition the image into the number of segments specified by the user. The segments produced have two desirable properties. First, the segments tend to be nearly square. A square region has a smaller perimeter than an elongated region of the same area, so its pixels have more neighbors with which correlations can be exploited and, thus, more effective compression is possible. Second, the segments tend to have nearly equal areas, which makes it somewhat more likely that compressed segments will have similar lengths; therefore, data losses are less likely to corrupt larger regions of the image. Figure 3 illustrates the partitioning of a square image into 8 segments.

Dividing an image into a large number of segments can confine the effects of a packet loss to a small area of the image; however, it’s generally harder to effectively compress smaller image segments. Since ICER provides flexibility in choosing the number of segments, compression effectiveness can be traded against packet loss protection, thereby accommodating different channel error rates. Note also that more segments are not always bad for compression effectiveness: many images are most effectively compressed using 4 to 6 segments because disparate regions of the image end up in different segments.

Because ICER is progressive, some error containment automatically occurs within segments as well: when a packet loss occurs, any previously received packets for the affected segment will allow a lower fidelity reconstruction of that segment, as illustrated in Figure 3.

Conclusion

With an armada of 18 cameras set to land on Mars in 2004, MER will rely heavily on ICER to deliver data back to Earth during the 180 Martian days of surface operations. Since ICER is a fairly general purpose algo-

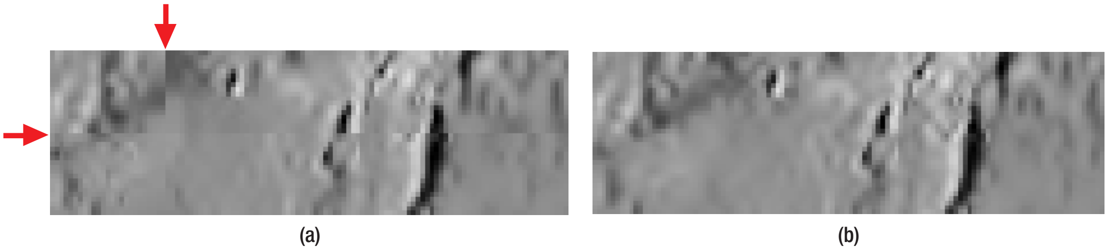

(a) (b)

Figure 2. Image details that illustrate (a) artifacts at segment borders that would arise if the wavelet transform were separately applied to each segment, and (b) elimination of such artifacts in ICER-compressed image

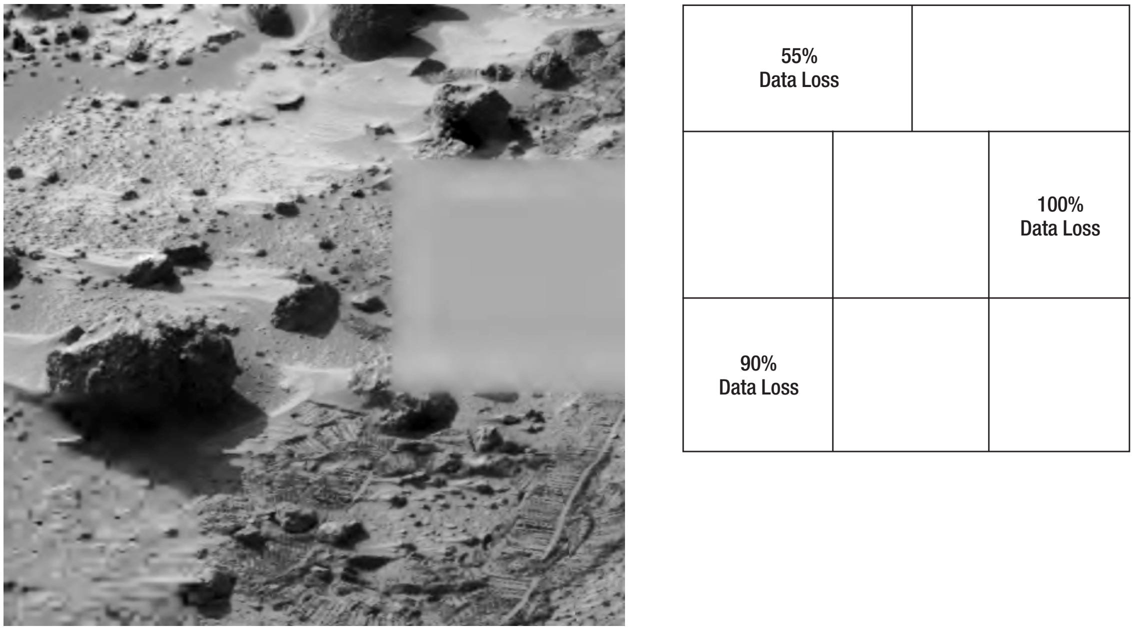


Figure 3. **Example of error containment in an image compressed to 1 bit/pixel, and suffering packet losses affecting three of eight image segments**

rithm and provides good lossy and lossless image compression combined with effective error containment, it has the potential to benefit other missions as well. Rapid advancements in imaging technology will continue to push the need for innovative data compression technologies such as ICER. As the techniques improve and hardware speeds increase, future missions will be able reap larger benefits from image compression.

References

[1] M. D. Adams and F. Kossentini, "Reversible Integer-to-Integer Wavelet Transforms for Image Compression: Performance Evaluation and Analysis," *IEEE Transactions on Image Processing*, Vol. 9, No. 7, pp. 1010–1024, June, 2000.

[2] P. G. Howard, "Interleaving Entropy Codes," *Proc. Compression and Complexity of Sequences* 1997, Salerno, Italy, pp. 45–55, 1998.

[3] A. Kiely and M. Klimesh, "A New Entropy Coding Technique for Data Compression," *IPN Progress Report* 42-146, pp. 1–48, April-June, 2001.

[4] A. Kiely and M. Klimesh, "Memory-Efficient Recursive Interleaved Entropy Coding," *IPN Progress Report* 42-146, pp. 1–14, April–June, 2001.

[5] A. Said and W. A. Pearlman, "An Image Multiresolution Representation for Lossless and Lossy Compression," *IEEE Transactions on Image Processing*, Vol. 9, No. 5, pp. 1303–1310, September, 1996.

[6] D. Taubman, "High Performance Scalable Image Compression with EBCOT," *IEEE Transactions on Image Processing*, Vol. 9, No. 7, pp. 1158–1170, July, 2000.

[7] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS," *IEEE Transactions on Image Processing*, Vol. 9, No. 8, pp. 1309–1324, August 2000.