Onboard Science Processing and Buffer Management for Intelligent Deep Space Communications[1]

Roberto Manduchi, Sam Dolinar, Fabrizio Pollara, and Adina Matache
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109
manduchi@telerobotics.jpl.nasa.gov, [sam,fabrizio,matache]@shannon.jpl.nasa.gov

Abstract—We present an integrated system for the intelligent progressive transmission of data for deep space communications. This work is motivated by the realization that much more information can be collected by imaging and remote sensing equipment than can be transmitted through downlink channels. Suitable onboard science processing allows us to introduce semantics to the data collected by the imaging and remote sensing equipment. The data stream is then prioritized according to its significance in the image, and the most significant segments of data are transmitted first by means of a prioritized buffer management strategy. We show that this system allows to optimally exploit the limited onboard resources (downlink data rate, buffer size) and therefore to maximize the science return of a mission.

TABLE OF CONTENTS

1. INTRODUCTION

Future spacecraft and Mars rovers will have the ability to acquire increasingly larger loads of scientific data. Imaging and multispectral instruments will be deployed in order to send to Earth as much information as possible, for fruition by both the scientific community and the general public. Unfortunately, the communication bit-rate represents an unavoidable bottleneck for the transmission of such loads of data. Traditional waveform compression techniques are plainly inadequate for this scenario, unless one is willing to accept a severe degradation of the entire image. In order to circumvent the communication bottleneck, one could in principle store images and other data in memory as they are acquired, for transmission during "low-activity" periods (for example, in the case of a Mars Rover, at night). However, the onboard buffer size is unavoidably limited, and part of the data needs to be thrown away. To makes things worse, in the case of deep space communications, interactivity is severely limited, due to the long latency and the scarce opportunities for uplink. Hence, "browsing" low-resolution versions of the images in order to decide which ones should be transmitted at full resolution (a technique widely used in multimedia communications on the Internet) is not an option here.

It is clear that the data gathered in a mission cannot be all transmitted to Earth. The question is: "What part of the data should be downlinked, and what should be trashed?" We will try to answer this question by:

1. Defining a metric for measuring the relative "importance" of gathered information;
2. Representing by a simple model the dynamics of data collection and the constraints and parameters of the processing/transmission system.

This approach will allow us to quantify the performance of any communication system by measuring the quantity of "important" information transmitted to Earth subject to the constraints of the model.

Clearly, traditional techniques for progressive compression (also known as scalable or hierarchical or embedded coding) can be analyzed under this framework too. For example, the so-called "spectral selection" method of JPEG for progressive coding [1] orders the compressed bitstream so that DCT coefficients corresponding to lower spatial frequencies in the image are transmitted first. This is equivalent to assigning more importance to lower spatial frequencies than to higher frequencies. More sophisticated techniques for embedded image coding such as the EZW algorithm [2] or the SPIHT algorithm [3] generate representations that are coarse-to-fine in both the spectral domain and range simultaneously. Thus, coefficients that correspond to coarser scales *and* have larger magnitude are transmitted first, thereby minimizing a more significant distortion criterion than simple spectral selection.

In general, progressive encoding represents an "optimal" design solution because it "adapts itself" to the available resources. A progressive encoder generates a properly ordered bitstream. If there is enough buffer memory to store all the encoded data, and enough time to transmit it, the entire image will be received. In general, these conditions will not be satisfied, and only a small fraction of the data will be transmitted. Although we may not be able to *a priori* estimate the size of the transmitted bitstream (since this is determined by the dynamics of the acquisition/processing/ transmission process), we are guaranteed that the "most important" information in the image will be transmitted.

The goal of this work was to extend the idea of progressive transmission by incorporating semantic value to the "importance" attributes of encoded data. In particular, we define a simple measure of *science return* based on information theoretic considerations as well as on the estimated *scientific value* of the transmitted data. The task of determining the relative scientific importance of segments of data is carried out by an *onboard science processing* module, designed according to guidelines provided by the remote user (the science community.) This module pre-processes the image and provides input to the progressive encoder in the form of a suitable "classification map". By combining the semantic characterization produced by the science processing module with the content-blind data organization criteria of traditional progressive encoder, we obtain a new measure of the "importance" of each segment of data. Algorithms that best utilize the available resources can thus be designed and their performances in terms of science return assessed.

Extensive studies are under way at JPL as well as at other NASA centers for the development of systems able to autonomously detect features of scientific interest in remotely acquired images. Many proposed techniques are based on computer vision and pattern recognition algorithms [4,5]. Increasingly available onboard computational power and memory are the key factors enabling a whole collection of science processing modules, targeted at various features of interest. The simplest embodiment of a science processing module is an algorithm that segments an image into "important" and "less important" areas. Clearly, the concept of "scientific importance" of a segment of data will change depending on the final user. For example, a geologist may be interested in finding carbonates on Mars, therefore she may consider as important any part of the scene characterized by the presence of layered geological formations. An astrobiologist may be interested in color bands on a rock surface, which may reveal biotic material. And, in general, any unexpected (and therefore highly informative) situation should probably be given highest priority. It is extremely important that the system be open to different interpretations of the data or, in other words, that it can accept a whole library of science processing modules, tuned to different science tasks.

We should stress the fact that while progressive encoders are content-blind, meaning that they know nothing about the scientific content of the images, science processing modules are representation-blind, in that they are unaware of the actual processing (such as transformations, quantization and entropy coding) performed by the encoder. Thus, the two modules can be logically decoupled. The interface between the science processing module and the progressive encoder is in the form of a classification map, which assigns each pixel to one class in a predefined set. The encoder uses this information to produce a hierarchical ordering of the compressed bitstream, which is both content and representation aware.

As discussed previously, besides providing a metric for the "importance" of a data segment, we need to be able to describe the dynamics of the acquisition/processing/ communication system from a global standpoint, and to optimize the system performance in terms of science return. It should be clear that, due to the possibly variable rate of data acquisition in deep space mission, as well as to the bit-rate bottleneck of the downlink channel, buffering is an extremely important operation. Buffer memory is a limited resource, and we should make the best use of it by means of a specific control strategy. The goal of the combined encoding/buffer system is to maximize the value of data transmitted and minimize the value of data lost due to buffer overflow. We introduce a control strategy that maximizes the buffer's usefulness by keeping it constantly full and overflowing. Simulation results show the excellent results achievable with our approach in terms of science return in dynamic environments.

2. Science Processing – A simple example

As discussed in the Introduction, the basic goal of a science processing module is to produce a spatial classification map for a given image. The class subdivision should be meaningful for the scientific task of interest, and contributes to determine the *priority* of the data segments to be transmitted. A whole library of science processing modules can be accommodated by the communication system, provided that they all use a common interface in the form of a classification map.

In general, it is not necessary that classification maps be defined just on the spatial domain. For example, one may characterize multispectral imagery not only by the image segments corresponding to features of interest but by the characterization of the multispectral features themselves (expressed in terms of the most relevant bands or perhaps by a suitable combination of bands). It should be pointed out that the determination of interesting areas in an image should depend on all available contextual information (for example, from the analysis of previously acquired images).

In order to provide a simple yet meaningful instance of science processing module, we have considered the scenario of a Rover traversing an arid territory with rocks scattered over an otherwise sandy area. While this should by no means be considered an actual emulation of a real Mars environment, it does give the flavor of the results attainable in a planetary exploration. The rocks that populate the sandy soil are of two kind: basalt (characterized by a reddish surface) and obsidian (with a blue/gray appearance). We assume that for this particular instance of science processing, a geologist is primarily interested in studying basaltic formations and, with a minor emphasis, the presence of obsidian. We also assume that sand is of minimal importance to the scientist, and that anything that cannot be characterized as basalt, obsidian or sand may reveal unexpected material and therefore should be given the highest transmission priority.

In our experiment, the science processing module performs a color-based classification of the image [6]. A mixture-of-Gaussians model [7] was used to represent the probability density function in RGB color space of each one of the M predefined classes. In other words, we model the conditional likelihood $p(\mathbf{c} \mid j)$ of any given 3-vector $\mathbf{c}$ representing an RGB tristimulus given a class j as follows:

$$p(\mathbf{c} \mid j) = \sum_{n=1}^{N(j)} \alpha_{n,j} g\left(\mathbf{c}, \mu_{n,j}, \Sigma_{n,j}\right) \tag{1}$$

where the coefficients $\alpha_{n,j}$ are the mixing parameters, $N(j)$ is the number of Gaussians in the mixture, and $\mu_{n,j}$ and $\Sigma_{n,j}$ are the mean vector and covariance matrix of the n-th Gaussian $g(\cdot,\cdot,\cdot)$ in the model:

$$g\left(\mathbf{c}, \mu_{n,j}, \Sigma_{n,j}\right) = \frac{1}{(2\pi)^{3/2}\sqrt{\det\left(\Sigma_{n,j}\right)}} e^{-\frac{1}{2}\left(\mathbf{c}-\mu_{n,j}\right)^T \Sigma_{n,j}^{-1}\left(\mathbf{c}-\mu_{n,j}\right)} \tag{2}$$

The model parameters ($\alpha_{n,j}$, $\mu_{n,j}$ and $\Sigma_{n,j}$) are estimated via Expectation Maximization [7] from a set of labeled training images. A Maximum Likelihood estimator initially classifies the test images by assigning each pixel $\mathbf{x}$ to the class j which maximizes the conditional likelihood $p(\mathbf{c}(\mathbf{x}) \mid j)$ (where $\mathbf{c}(\mathbf{x})$ is the color of pixel $\mathbf{x}$). Spatial coherence is then enforced by means of Besag's Iterated Conditional Mode algorithm [8]. The determination of *outliers*, that is, of features that cannot be explained by any of the M models, is operated by setting a threshold p_{ol} on the unconditional likelihood $p(\mathbf{c})$, defined by

$$p(\mathbf{c}(\mathbf{x})) = \sum_{j=1}^{M} P(j) p(\mathbf{c}(\mathbf{x}) \mid j) \tag{3}$$

where $P(j)$ is the prior probability of class j (in our experiments, we set $P(j)=1/M$ for all classes). Thus, all pixels with color $\mathbf{c}$ such that $p(\mathbf{c})<p_{ol}$ are classified as outliers.

The classes considered in our experiments (see Figures 4 and 5) are the "Basalt" class (represented in green), the "Obsidian" class (represented in yellow), the "Sand" class (represented in gray), the "Shadow" class (represented in brown) and the "Outlier" class (represented in red). The "shadow" class was introduced after the experimental observation that image areas in dark shadow cannot be classified robustly. In this case, classification into any of the other classes cannot be trusted. Note that it would be incorrect to classify such point as outliers: indeed, outliers are a consequence of model inadequacy and not of unreliable discrimination.

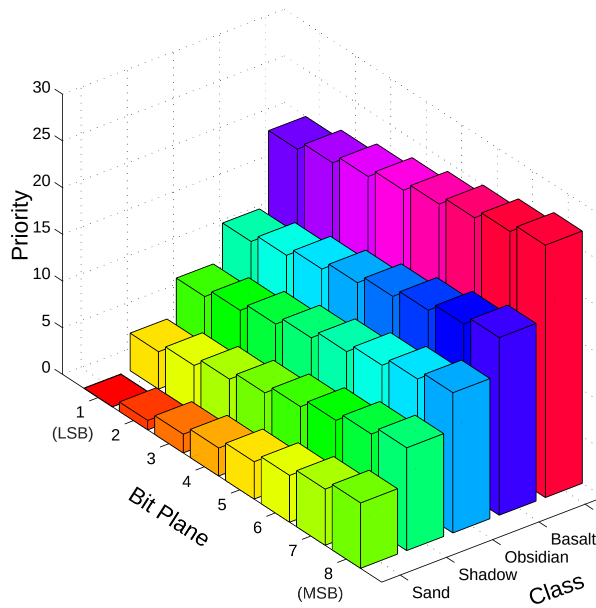
The classification procedure described above assigns each image pixel to one class. For implementation reasons that will be clear in Section 4, it is useful to consider also a coarser granularity, by assigning a single class label to each one of the 8×8 pixel groups tiling the image plane in a regular lattice. The block assignment can be carried out in a number of ways. We have adopted a strategy of weighted majority voting: the 8×8 block under examination is assigned to the class j which maximizes the quantity $w_j n_j$, where n_j is the number of pixels in the block assigned to the class j, and w_j is a weight which represents the "importance" of the class in this context.

3. Assigning Priorities

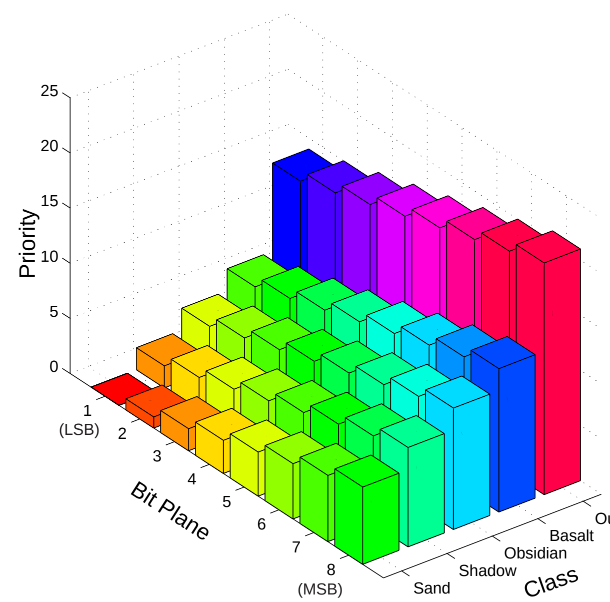
The role of an encoder is to produce a compact representation of the data, for example by minimizing the average symbol length (entropy coding). Additionally, a *progressive* encoder should order the data in such a way that the most "important" segments of the data are transmitted first. Bit-plane ordering provides perhaps the simplest instance of progressive encoding. Assume for the time being that we are dealing with monochromatic images. Bit-plane ordering organizes the encoded bistream in such a way that the Most Significant Bits (MSB) of all pixels in the image are transmitted first, followed by the second most significant bits of all pixels, and so on. If only a part of the bitstream is transmitted, a coarse version of the original image is received; additional bits will contribute to enhance the received image. In the case of bit-plane ordering, the concept of "importance" has a clear Signal-to-Noise Ratio counterpart (indeed, each bit contributes to around 6 dB of quantization SNR).

For the case of color images considered in our examples, we slightly modify the bit-plane ordering algorithm to account for the intrinsic vectorial representation of the color data. The RGB tristimuli are first converted into YCrCb tristimuli by means of a suitable linear transformation [9]. We assume that each component of a YCrCb tristimulus is quantized with 8 bits. The "Y" component of a tristimulus represents the pixel luminance and assumes only positive values, while the "Cr" and "Cb" components (which represent the chrominance channels) can take on both positive and negative values, and therefore can be represented by one bit for the sign and seven bits for the magnitude. To encode the color values, we use a "generalized" bit-plane representation: each atomic element is a triplet of bits, one from each component of the tristimulus. In particular, the "most significan triplet" is composed by the MSB of the luminance and by the sign bits of the two chrominance channels. The "second most significant triplet" is composed by the second most significant bit of the luminance and by the most significant bit of the 7-bit representation of the chrominance magnitudes, and so on. For simplicity's sake, we will still use the terminology "bit plane ordering" in the following, although the reader should mentally substitute the word "triplet" for "bit".

To keep science value into account, we should modify the simple notion of "importance" by considering a combination of bit-plane index and classification value. In other words, the MSBs as well as some other bit planes of the most "scientifically important" image segments should be transmitted before any piece of data from image areas belonging to other classes. For instance, in our example, we may want to transmit, say, the first 4 bit planes of pixels belonging to the class "Basalt" before transmitting any bits for the classes "Obsidian", "Shadow" or "Sand". This can be achieved by setting a *priority* value to the each bit of the data stream, obtained by suitably combining the "class" index and the bit-plane index. Two possible priority assignments relative to our experiment are shown in Figure 1 (a) and (b). The difference between the priority functions of Figure 1(a) and (b) is in the "softness" of priority assignment among the classes "Sand", "Shadow", Obsidian" and "Basalt". For example, in both cases the most significant bit of a pixel belonging to the "Obsidian" class has higher priority than the least significant bit (LSB) of a pixel belonging to the "Basalt" class, since we don't want to wait for all the data coming from "Basalt" areas before receiving information about the "Obsidian" ones. More specifically, the MSB of a pixel belonging to the "Obsidian" class has the same priority as the fourth most significant bit of a pixel belonging to the "Basalt" class in the case of Figure 1(a), and of its second most significant bit in the case of Figure 1(b). However, all bits of any pixels classified as "Outlier" have higher priority values than the rest, since we assume that any "unexpected" feature is "infinitely more interesting" than anything we may have already considered. The choice of priority function should be based on 1) the confidence we have on the science classification and 2) the relative importance of different science classes to the final user.



(a)



(b)

Figure 1: Two possible priority assignments as a function of science class and bit-plane index.

Thus, it is these "priority" values that represent the overall importance of a bit of information. It should be clear that our definition of priority is fairly arbitrary; it depends on: a) the science processing module and the "scientific priority" assigned to the classes; b) the signal representation, which may be extremely more complex than the simple bit-plane ordering described above; c) the combination rule between science classes and pixel representation. Given a particular definition of priority, the sum of the bit priorities of the data transmitted in a mission will be given the evocative name of *science return*. The optimal communication system is thus the one that maximizes the science return in a mission, given the overall data rate and memory constraints. We will discuss in the next section a technique that ensures a high rate of science return by means of a very simple buffer management algorithm.

In closing this section, we should point out that in our experiment the data is not actually "compressed", in the sense that, to receive an image in its entirety, we need to transmit as many bits as the original raw data. (Actually, additional bits are transmitted in the form of "metadata", as explained in the next section). The design of suitable forms of data compression will be the goal of future work. Nevertheless, the results presented here show that by simply adopting a suitable methodology of prioritized buffer management, the science return can be significantly increased with respect to traditional content-blind techniques.

4. PRIORITIZED BUFFER MANAGEMENT

A buffer is required every time there is disparity between the encoded data rate and the downlink transmission rate. The buffer can accept, output, and trash segments of data. A "data segment" (or "buffer packet") is the smallest data unit handled by the buffer. In principle, a data segment could be a single bit. However, for the purpose of the present work, a data segments is the set of bits corresponding to a given bit-plane of an 8×8 image block. The reason for choosing such a coarse granularity resides in the fact that, due to the prioritized buffer management described in the following, the decoder, upon reception of a segment of data, cannot autonomously infer its location in the image. Thus, some additional information, in the form of "metadata", needs to be transmitted with each data segment. By using larger granularity, we can limit the relative size of the metadata. However, it is easy to convince oneself that finer granularity ensures a more efficient utilization of the buffer at the expense of processing load.

Our buffer is organized according to the following rules:

1. The buffer is an ordered list of data segments, whose ends are named *head* and *tail* respectively. A partial order is established in the list, and is determined by the priority of the data segments (as defined in the previous section). In other words, suppose that a segment s_1 in the list has higher priority than another segment s_2 in the list. Then s_1 is always positioned closest to the head than s_2 (we will say that s_1 is placed *above* s_2 and therefore s_2 is placed *below* s_1).
2. Data segments are output from the head of the buffer only. The output rate is constant (indicated by R_{out}), unless the buffer is empty, in which case R_{out} =0.
3. An input data segment will always be accommodated in the buffer unless the buffer is full of higher priority segments. The data segment is placed below any other segment with equal or higher priority. Once a data segment is in the buffer, its position with respect to the other segments never changes.
4. If the buffer is full when the current input data segment is available, the current input segment will be accommodated only if there is some other segment with lower priority (in which case this segment will be discarded). If all segments in the buffer have priority equal to or higher than the current input segment, the current input segment is discarded.

An example of buffer evolution is shown in Figure 2. Only two priorities are considered here: the lower priority segments are represented by the letter "A" and light blue color, while the higher priority segments are labeled by the letter "B" and colored in dark blue. The number on each segment represents the corresponding input order. Note that, due to the limited buffer size and to the insufficient output rate, the segment of index 5 has to be discarded. It is not by chance that a segment of lower priority is discarded: our buffer management strategy ensures that only segments of the lowest priority in the buffer are discarded, and that only segments of the highest priority in the buffer are output. In general, when a very high-priority data segment arrives at the buffer, it will be output shortly, while a very low-priority segment will be soon trashed. Segments of average priority usually spend much more time in a "limbo" situation inside the buffer; whether they will eventually be output or discarded depends on the priority characteristics of the future data presented to the buffer. A discussion of the "discriminatory power" of a prioritized buffer is presented in Section 5.

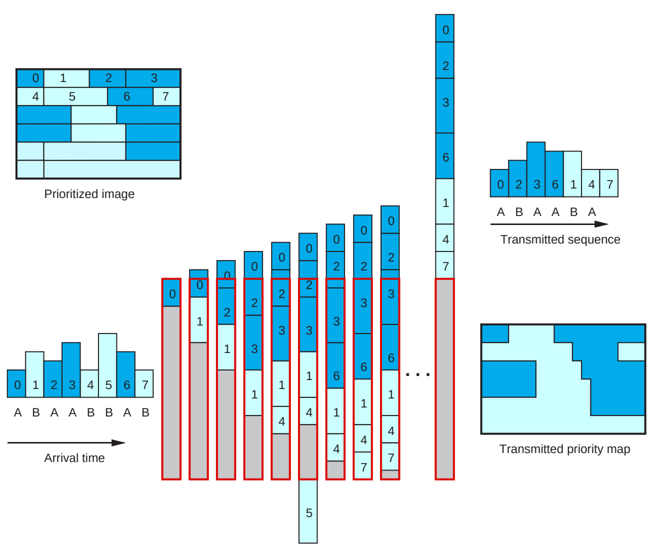


Figure 2: Evolution of a prioritized buffer.

Another possible implementation of the buffer is shown in Figure 3. This implementation uses separate buffers for each priority. This method can be used to emulate the previous method if the sum of the individual buffer sizes is equal to the size of the entire buffer.

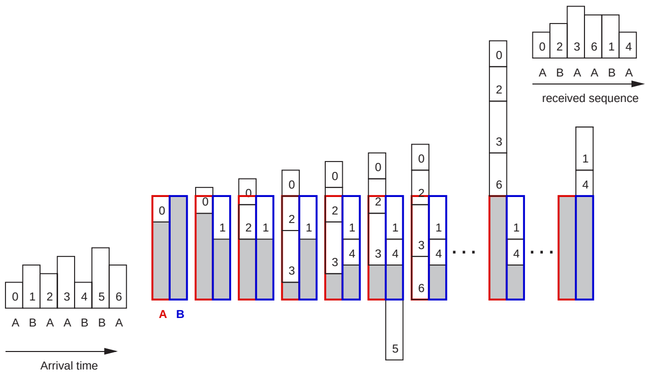


Figure 3: Evolution of a prioritized buffer with separate buffers for each priority.

Concrete examples of the improved performance in terms of science return attainable by the prioritized buffer strategy are shown in Figures 4 and 5. The data is collected and presented to the buffer according to the scenario described in Section 2. In particular, the images were taken in a sequence: each image was acquired only after the previous image had been processed and input to the buffer. Once an image was acquired, it was first color classified (as represented in the second row), and the priorities of each bit (triplet) plane for each 8×8 block were determined. Then, the image was blockwise scanned in raster order, and the bit (triplet) planes for each block were sequentially input into the buffer. The data relative to a block was input in the buffer only after all the bit-planes in the previously scanned block were processed. It was assumed that the rate of the data into the buffer (R_{in}) was 5 times the output rate R_{out}, and that the size of the buffer was 16 times smaller than the size of an image. (A noticeably undersized buffer was chosen in order to demonstrate the relevant dynamics of the buffer scheme without the need for a large-scale simulation). In the third row we show the decoded data output from the buffer. As a comparison, in the fourth row we show the result assuming that no science classification was performed, i.e., the priority of a bit-plane is determined solely by its index. In this case, the encoder knows nothing about the *semantics* of the image, and therefore puts the same effort to transmit, for example, bits belonging to the "Sand" class and to the "Basalt" class. The difference between the two cases in terms of science return is appreciable by simple visual inspection; it is quantified by the numbers in the small colored cells at the upper left corner of the figures. These are the numbers of data segments belonging to each class for each original image (second row) and for each transmitted image (third and fourth row). It can be seen that very few (if any) data segments corresponding to the "Sand" class are transmitted with our prioritized strategy, while many more data segment belonging to the "Basalt" class are transmitted than in the science-unaware case. In particular, note that all data segments corresponding to the "Outlier" class are transmitted in the science-prioritized case.

5. Performance Measures

First we consider a case in which all data segments are of equal size. Assume that no more than K data segments can be transmitted during a given period of time T. We make several conceptual definitions in order to quantify the performance of the onboard science processing and buffer management system.

An *ideal ground-based scientist* designs experiments to collect and transmit exactly the K data segments deemed most valuable based on *a priori* information available to the scientist on the ground before any data is actually collected by the spacecraft. An *ideal onboard scientist* designs experiments to collect a lot more than K segments on the expectation that many potentially valuable discoveries cannot be anticipated from *a priori* information. Later, onboard the spacecraft, this mythical scientist reviews the data collected and culls the K most valuable segments for transmission to earth. This *a posteriori* determination of value is based not only on the *a priori* information available to the ground-based scientist, but also on the *content* of the data actually collected.

These two paradigms serve as useful performance benchmarks because the goal of the onboard science processing and buffer management system is to approximate the selectivity of the ideal onboard scientist. Measured against the paradigm of the ideal onboard scientist, the automated onboard system will always fall short. However, if the performance gap between the ideal ground-based and onboard scientists is large, any decent approximation by the automated system to the selectivity of the ideal onboard scientist will still accomplish large gains in total data value returned. As an example, suppose that, by going onboard instead of staying on the ground, an ideal scientist could quadruple the total value of data returned by peeking at the content of the data before committing scarce downlink resources. Then, if the automated onboard system performs only half as well as the ideal onboard scientist, it would still double the total data value returned by the most carefully designed experiments of any ground-based scientist.

Thus the performance assessment of the automated system can be conceptually separated into two components: first, the (potentially huge) *gains* obtainable by substituting an ideal onboard scientist for an ideal ground-based scientist, and, second, the (hopefully small) *losses* suffered by substituting the automated onboard system for an ideal onboard scientist. The first of these two components: (a) is guaranteed to be a nonnegative gain, because the onboard scientist can always choose to ignore the content of the data

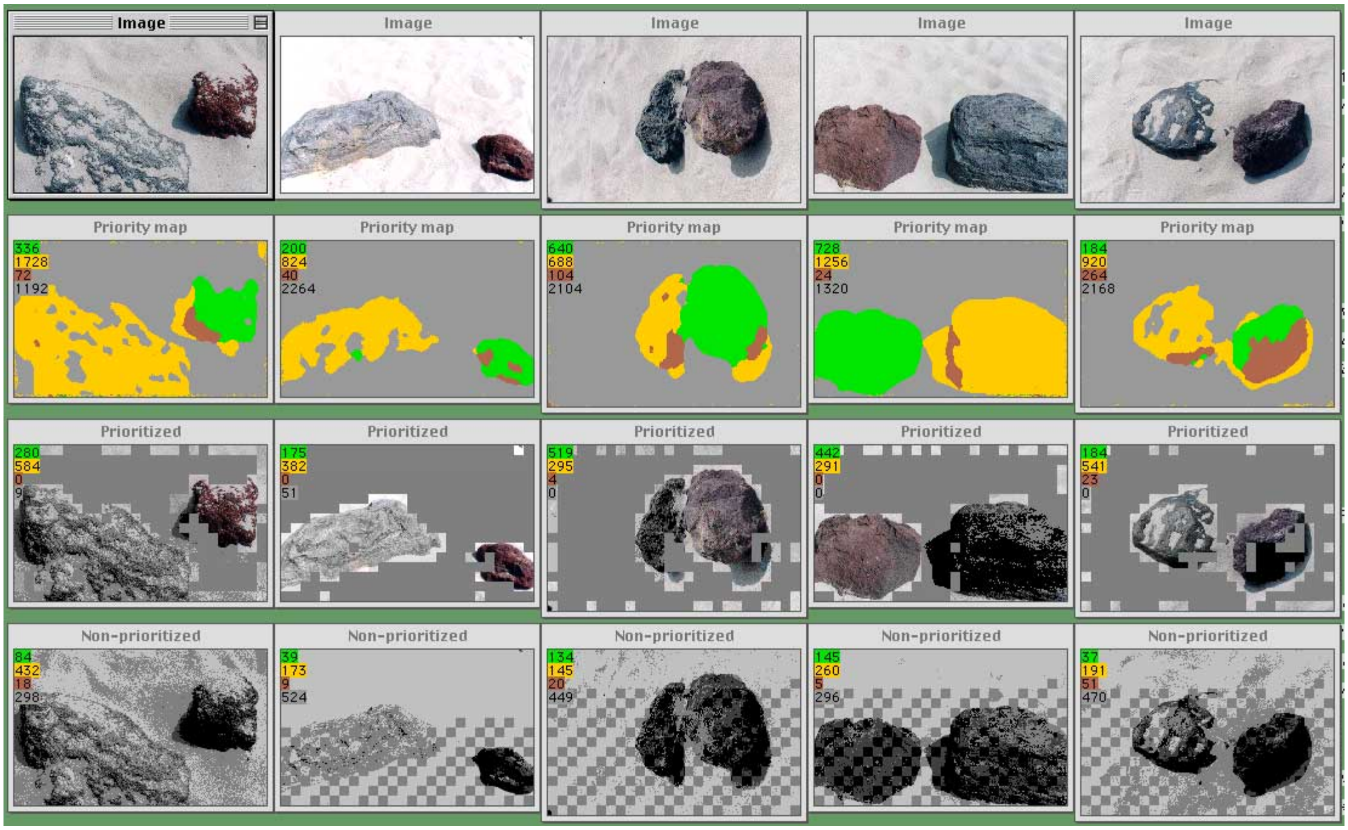


Figure 4: Simulation results. Top row: original images. Second row: classification maps. Third row: Data transmitted with prioritized buffer. Fourth row: data transmitted without science prioritization. Top left corners: number of data segments belonging to each class for each original image (second row) and for each transmitted image (third and fourth row). The prioritization function of Figure 1(a) was used. The size of the buffer was 16 times smaller than the size of one original image. The output rate R_{out} was 5 times smaller than the input buffer rate R_{in}.

and transmit exactly the same data as the ground-based scientist, and (b) is independent of the onboard system and therefore can be estimated entirely within the scientific community without any reference to the particular onboard system implemented. This conceptual separation of performance components allows us to focus our attention on measuring how well the automated onboard system performs relative to what it is trying to approximate: by how much does the automated system fall short in mimicking the decision-making and data-handling of the ideal onboard scientist? This performance component itself consists of several sub-components reflecting on the goodness of various pieces of the full onboard system. The most important of these is how well does the onboard science processor duplicates the classifications and valuations of the ideal onboard scientist. But, even if these valuations were exactly the same, the onboard prioritized buffer manager would still not select exactly the same K segments for transmission, due to finite limits on processing power and buffer space.

To evaluate the performance of the onboard buffer manager in isolation from that of the onboard science processor, we define the *discriminatory power* of the onboard buffer manager in terms of its ability to process for transmission the same K segments that are deemed most important by either an ideal onboard scientist or an automated onboard science processing system. A discriminatory power of 1 means that exactly the same K segments are selected for transmission, while a discriminatory power less than 1 measures the ratio of the total value of the K segments transmitted to the total value of the K segments deemed most important.

Here are some of the practical system considerations that can cause the buffer manager's discriminatory power to be lower than 1. First, if the size of the buffer is very small, then statistical fluctuations in the value of data flowing into the buffer can temporarily overload the buffer with high-value data, some of which must necessarily be lost, and at other times flood the buffer with low-value data, some of which gets transmitted because nothing better is available at the time. This problem can be greatly exacerbated if the statistics of the collected data are dramatically non-stationary in time. The greater the non-stationarity, the bigger are the buffers required to average effectively over collection periods of low-value and high-value data.

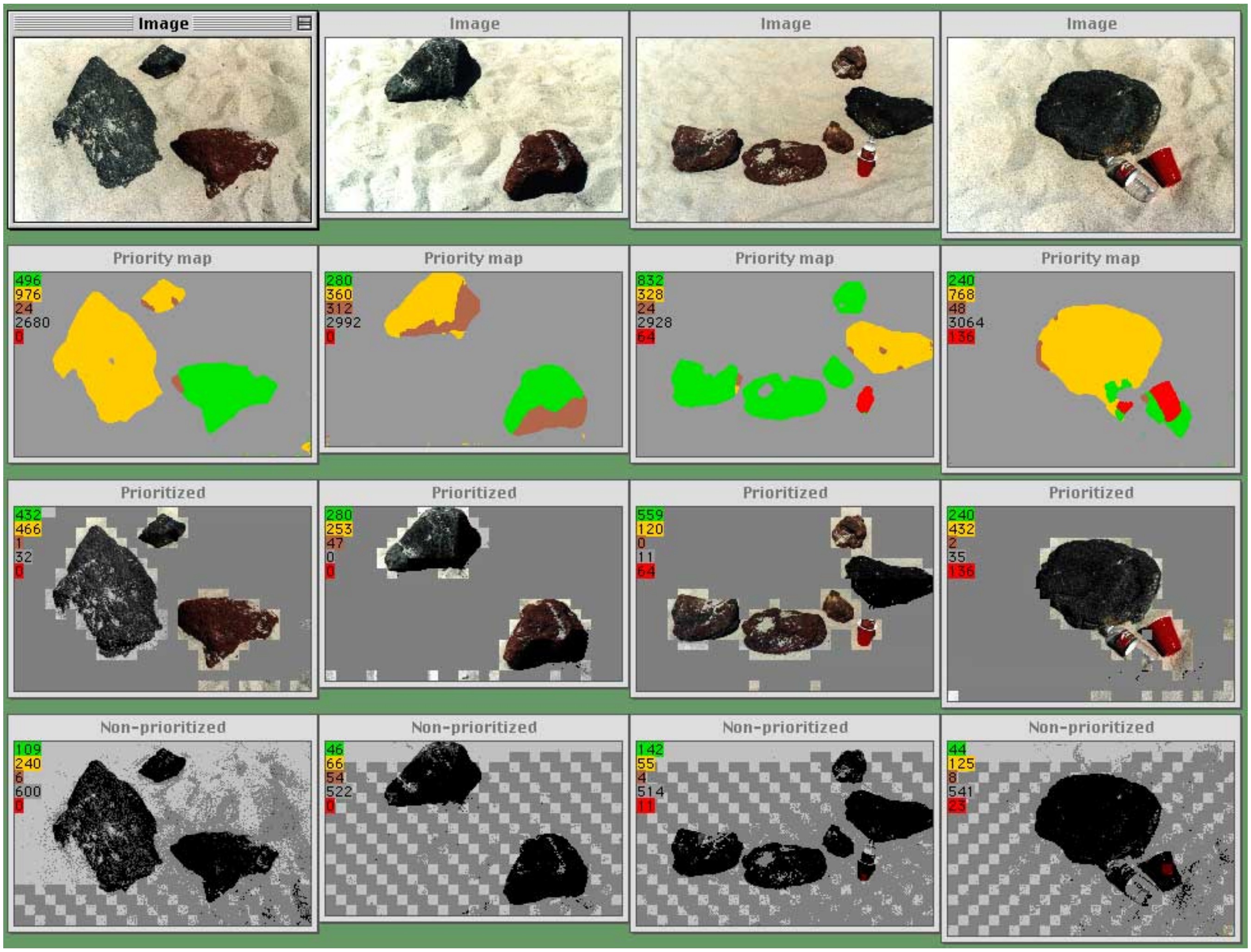


Figure 5: See caption of Figure 4.

Second, finite onboard processing power will limit the amount of sorting and data-handling affordable to the buffer manager, and thus the best *K* segments might not be selected for transmission even if the buffer size were infinite and no segments ever had to be discarded. Third, the prioritized handling of buffer segments can require a significant amount of “metadata” to be transmitted along with the raw data, to instruct the ground processor how to resort and reassemble the received segments. Given a fixed downlink constraint, the requirement to send metadata reduces the number of transmittable segments to something lower than *K*. This overhead due to metadata increases as the granularity of the classifications produced by the onboard science processor and the priorities handled by the buffer manager become finer and finer.

To analyze how well the prioritized buffer management algorithms performed in our experiment, we gathered some statistics that give a more detailed look at the buffer manager's discriminatory power than our previous definition of discriminatory power as a simple ratio. No "Outlier" class was considered in this experiment. Figure 6 shows histograms of the incremental science values of buffer packets transmitted, lost, or remaining in the buffer at the end of the experiment. The three histograms are almost completely non-intersecting. In this test, all packets with values 15 through 20 were transmitted, along with most of the 14's and about half of the 13's. At the other end of the scale, all packets with values 1 through 6 were discarded, along with about half of the 7's. All packets with values 8 through 12 were in limbo, still occupying the buffer, at the end of the experiment. Up to this point in the data transmission process, the only improvement to be offered by an ideal onboard scientist over the automated buffer manager's performance (assuming the same assignment of segment values and the same downlink constraint) would be the earlier shipping of the few value-14 packets still remaining in the buffer, leaving some extra untransmitted value-13 packets in the buffer in their place. Except for this one small difference, the ideal onboard scientist would have selected exactly the same packets for transmission as the automated buffer manager. Then, depending on the values of the data arriving after the experiment was ended, the value-14 or value-13 packets remaining in the buffer would eventually either get transmitted anyway or else get discarded if the buffer were suddenly overwhelmed with higher-value packets. Only in this latter case would there be any difference at all between the total values transmitted by the ideal onboard scientist and the automated buffer manager, and in this case the value difference would be just 1 value-unit multiplied by the number of value-14 segments that are ultimately discarded due to the slight non-optimality of the buffer manager's operations.

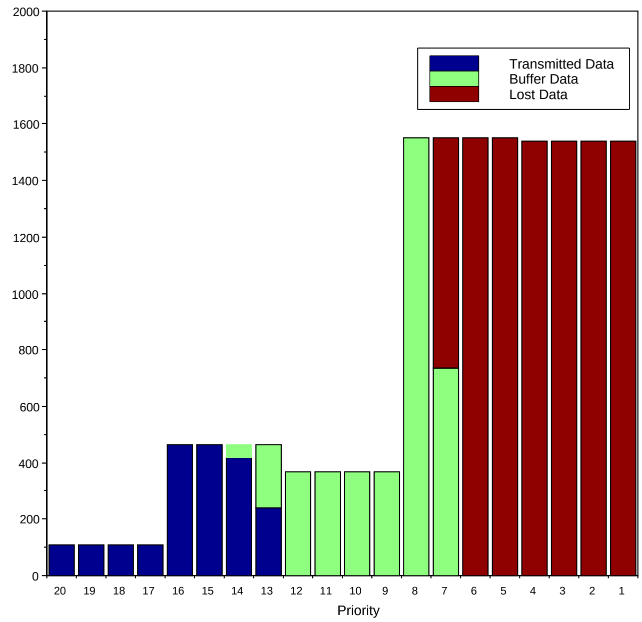


Figure 6: Histogram of transmitted, buffered, and lost data. R_{in}=6R_{out}. The size of the buffer and of the image were the same, and the prioritization function of Figure 1(a) was used. (16×16 blocks)

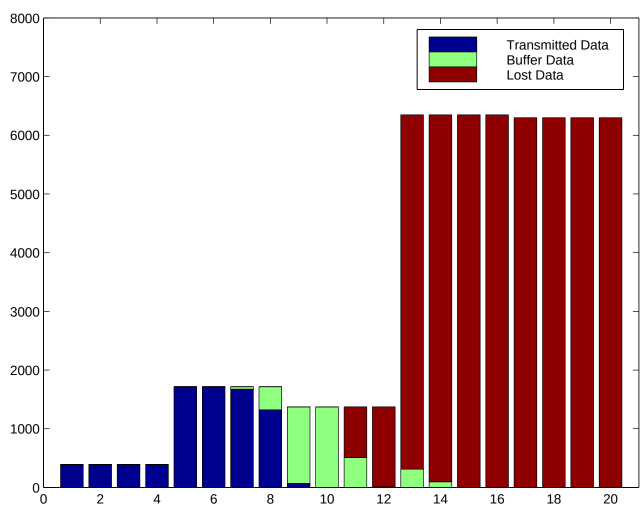


Figure 7: Histogram of transmitted, buffered, and lost data. R_{in}=6R_{out}. The size of the buffer was 4 times less than the size of the image, and the prioritization function of Figure 1(a) was used. (8×8 blocks)

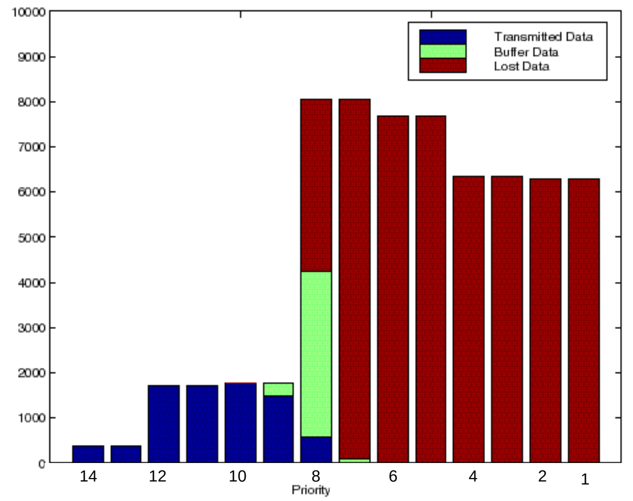


Figure 8: Histogram of transmitted, buffered, and lost data. R_{in}=6R_{out}. The size of the buffer was 4 times less than the size of the image, and the prioritization function of Figure 1(b) was used. (8×8 blocks)

Up to now, we have assumed for simplicity that all data segments are of equal size. In actuality, the data segments produced by a good progressive compression algorithm are highly variable and often unpredictable in size. It is easy to generalize the paradigms of the ideal ground-based scientist and the ideal onboard scientist to the case of variable data segment sizes. Now, instead of facing a fixed budget of K segments, the goal of each type of scientist is simply to maximize the total value of a set of K' segments with sizes that together do not exceed a fixed total downlink budget of m bits (over a given period of time T)

The variable compressibility of data segments greatly widens the benchmark performance gap between the ideal onboard and ground-based scientists. Whereas the ground-based scientist can only guess at the downlink load demanded by a given data segment, the onboard scientist can optimize the value returned *per bit* of data transmitted, thus optimizing the allocation of downlink resources. This optimization of value per bit transmitted can also be accomplished with good efficiency by an automated onboard system (having sufficient computational and memory resources), because the downlink rate constraint and the relevant data segment sizes are simple numerical quantities that are known to the onboard buffer manager. Thus, we expect the automated onboard processor to reap a high proportion of the extra gain achievable by an ideal onboard scientist due to having knowledge of variable compressed data sizes that are not available to the ideal ground-based scientist. The overall efficiency of the onboard buffer manager in approximating the selectivity of the ideal onboard scientist can be measured by using a slight generalization of the discriminatory power ratio defined earlier for fixed segment sizes, namely the ratio of the total value of the K'' segments selected for transmission by the onboard buffer manager to the total value of the K' segments deemed most important by the ideal onboard scientist, with both sets of K'' or K' segments satisfying the downlink constraint on the total number of bits transmittable.

6. Conclusions

We have presented a scheme for progressive data transmission that incorporates science-based priority to the data processing. The limited resources of a spacecraft (in terms of downlink data rate and buffer size) can be optimally exploited by means of our prioritized buffer management scheme, thereby maximizing the science return of a mission. One of the results of our analysis of the prioritized buffer management scheme is that "virtually ideal" results can be achieved with modest-sized buffers (at least compared to large data storage buffers already needed onboard for other reasons such as waiting for the next station pass). An "ideal buffer manager", facing a collect-to-transmit ratio of X, would allow exactly the most valuable fraction $1/X$ of the total (compressed, in general) data volume to be transmitted. Although an ideal buffer manager would in principle require an infinite buffer, we demonstrate that very modest-sized buffers are sufficient to "discriminate" almost ideally between the valuable data segments that should be transmitted and the less valuable ones that should be trashed. Thus, we are not anticipating that the intelligent buffer management system will add significantly to existing onboard storage requirements.

Future research will be devoted to elaborating on existing schemes for Region-of-Interest coding [10], in order to effectively compress the prioritized data and to minimize the need for metadata. Error resilience schemes will also be developed for robust transmission in the prioritized case.

Finally, we would like to point out that our onboard science processing and prioritized buffer management system is envisioned as a tool that scientists can utilize directly in planning their data collection efforts, not just as a "black box" that processes their data after they have gathered it. If scientists gain confidence in this tool they will be able to design their experiments to deliberately collect far more data than could ever fit through the downlink, because the onboard system will keep the downlink busy with the best data collected. So this system is not at the stage of trying to gain "science community acceptance of the results", rather we are at the stage of demonstrating to scientists a potentially powerful tool that will enable them to collect and select the data they are most interested in. This will probably lead to custom design of "front end" science processing and/or data compression modules for different science applications (with the corresponding scientists intimately involved in developing the science processing algorithms), but with the commonality of being handled by the same priority-based data management system.

Acknowledgments

The work described in this paper was funded by the TMOD Technology Program and performed at the Jet Propulsion Laboratory, California Institute of Technology under contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology. John Wright and Eric Mjolsness are acknowledged for fruitful discussions.

References

[1] William B. Pennebaker and Joan L. Mitchell, *JPEG: Still Image Data Compression Standards*, Van Nostrand Reinhold, New York, 1992.

[2] Jerome M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", *IEEE Transactions on Signal Processing* ***4****(2), 3445-3462*, December 1993.

[3] Amir Said and William A. Pearlman, "A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", *IEEE Transactions on Circuits and Systems for Video Technology* ***6****(3), 243-250*, June 1996.

[4] Tara A. Estlin, Alexander G. Gray, Tobias P. Mann, Gregg Rabideau, Rebecca Castano, Steve Chien, and Eric D. Mjolsness, "An Integrated System for Multi-Rover Scientific Exploration", accepted for presentation at *AAAI-99*, 1999.

[5] V.C. Gulick, R.L. Morris, M. Ruzon, and T.L. Roush, "Autonomous Science Analyses of Digital Images for Mars Sample Return and Beyond", *30th Lunar and Planetary Science Conference*, March 1999.

[6] Glenn E. Healey, Staven A. Shafer, Lawrence B. Wolff, (eds.) *Color*, Jones and Bartlett Publishers International, London, England, 1992.

[7] Cristopher M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, England, 1995.

[8] Julian Besag, "On the Statistical Analysis of Dirty Pictures", *Journal of the Royal Statistical Society B,* ***48****(3), 259-302*, 1986.

[9] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley Publishing Company, 1997.

[10] David Nister and Charilaos Christopoulos, "Lossless Region of Interest with a Naturally Progressive Still Image Coding Algorithm, *Proceedings of the 1998 International Conference on Image Processing, 856-860*, 1998.

Roberto Manduchi received the Doctoral Degree in Electrical Engineering from the University of Padova, Italy. He worked as a Post-Doctoral Fellow at the University of California, Berkeley, at Caltech, and at Stanford University, conducting research on computer vision and signal processing. He later worked for Apple Computer, Inc., in the QuickTime VR group. He is currently a member of the technical staff at the Jet Propulsion Laboratory, in the Machine Vision and Tracking Sensors Group.

Samuel Dolinar is a member of the Technical Staff in the Communications Systems and Research Section, JPL. He received S.B., S.M., E.E., and Ph.D. degrees MIT. He came to JPL in 1980 after four years at MIT Lincoln Laboratory. At JPL, Dr. Dolinar initially worked on deep-space optical communication systems, electronic warfare software, and tactical communication systems. During the past fourteen years, his work has focused on channel coding and source coding for the deep-space channel. He has helped to develop improved bounds on the performance of linear block codes and on the complexity of trellis. During the past five years, his work has focused on turbo codes and turbo-like codes, and their associated iterative decoding algorithms. He taught a graduate-level course on Data Compression as a part-time Lecturer in the Electrical Engineering department at Caltech.

Fabrizio Pollara is a Technical Group Supervisor in the Communications Systems and Research Section, JPL. He received M.S and Ph.D. degrees in Electrical Engineering from U.C.L.A. Since 1983 he has been with JPL, where he contributes to the development of advanced communication systems for deep space missions. Research work includes development of new error-correcting coding systems. Dr. Pollara was responsible for initiating research on advanced data compression at JPL, which was a key component in the effort to improve Galileo's data return. He has taught error-correction coding theory as a part-time Lecturer in the Electrical Engineering department at the Caltech. Before coming to the U.S. with a Fulbright scholarship, he received a doctoral degree in Electrical Engineering from the Politecnico of Milano, Italy.

Adina Matache is a member of the Technical Staff in the Communications Systems and Research Section, Jet Propulsion Laboratory, Pasadena. She received the B.S. (1995) and M.S. (1997) degrees in Electrical Engineering from University of Washington. Since 1997 she has been working at JPL, where she has contributed to the development of error-correcting coding systems and data compression algorithms for deep-space missions. Currently she is pursuing a Ph.D. degree in Electrical Engineering at the University of California, Los Angeles.