VLA COMPUTER MEMO NO. 173

TRANSMISSION OF DIGITAL IMAGES OVER SLOW COMMUNICATION LINES

Arnold Rots

17 October 1984

It seems likely that in the not too distant future the N.R.A.O. will enter into data reduction modes where the bulk of the number crunching will be done at a geographical location different from that of the user. There are no conceptual problems with this, except in cases where the user wants to monitor the data reduction process and only has a slow data communication line available to the processing center. In many instances this presents no problem: the processing computer can be asked to provide summaries of the data (e.g., statistics over selected map areas to judge the progress of image restoration techniques). However, it is unavoidable that from time to time entire map(subsections) have to be sent over. Just for the sake of argument, sending a 4096 16 bit map over a 1200 baud line takes about 80 hours; it is stretching one's faith in hardware a bit too far to assume that both computers, plus the communication link, will stay up that long, not to mention the user's patience. Now obviously no one would want to send over the entire map in such a case and one may be able to increase the baud rate somewhat, but unless one goes to dedicated high capacity lines one is not going to bring the numbers down to a realistic level.

In this memo I will describe three techniques that are useful to bring the problem down to realistic proportions. I will also give the results of some experiments I have done. The techniques are progressive transmission, histogram equalization, and bit compression.

1. TRANSMISSION FORMAT

It is not necessarily true that transmission of the naked bits (however they are encoded) is the most efficient way of transmitting large quantities of data. Communication channels are not perfect and do require some error checking. To further be compatible with file formats acceptable to KERMIT, I shall assume in the following that the data are transmitted as ASCII files with 80 characters per record; or, 7 bits of data per 8-bit byte with a parity bit. There is nothing special, however, about the 7 bits per byte or even the records: after taking out the parity bits, the information is considered to exist of one single bit string, to be broken up by rules that have nothing to do with bytes or records. The only exception to this may be the first (few) record(s). These may contain formatted ASCII information on the size of the image being transmitted, the number of bits that it started from, and the top of the pyramid (see section 2).

2. PROGRESSIVE TRANSMISSION

Often a general impression of what the map looks like suffices for the user's purposes, although sometimes he may need the full detail on a subsection as well. The problem with conventional image transmission (line-by-line) is that it takes quite a while to build up enough of the image to be able to judge the whole. Progressive transmission schemes start with a very fuzzy representation of the image and then proceed to increase the sharpness of it. Once the user has enough resolution, he can stop the transmission, or decide to increase the resolution only in a particular subsection. This allows for considerable savings in transmission time and is highly interactive (assuming that a display program has access to the image as it is being received and decoded). A popular scheme is to build up a pyramid of images: at the bottom is the original map, the next one consists of the sums of all 2x2 pixel squares of the original map (and is therefore one quarter its size), and so on, until one gets to the top of the pyramid, consisting of a single number, the sum over the entire map. Transmission starts from the top of the pyramid. Once one has a particular level of the pyramid, the next level down can be reconstructed by providing three numbers per 2x2 square which, together with the sum contained in the upper level map, allows one to retrieve all four numbers in each square. Thus, the total number of numbers in a complete transmission is the same as for the original map, while one has the advantage that a complete, albeit blurred, image is available right from the start and transmission can be stopped at any point in the pyramid without having to make a priori assumptions about the resolution desired by the user. Since we are dealing with sums, the number of bits per number is higher, but this will not be more than one bit.

3. HISTOGRAM EQUALIZATION

In 16-bit maps, most of the bits are, on the whole, wasted, even assuming that the dynamic range is higher than 20dB. Usually, most of the higher levels are not filled. An good way of making efficient use of the bits is by histogram equalization: an intensity transformation is performed such that in the resultant map there are approximately equal numbers of pixels at each intensity level. This enables the user usually to compress a 16-bit high dynamic range map into 8 bits without appreciable degradation: a savings of a factor two, although one does have to send a look-up table along.

4. BIT COMPRESSION

In most maps the differences between adjacent pixels are considerably less than the number of bits in which the whole map is encoded - not only because of the finite beam size, but also because most maps are predominantly empty. It seems that one ought to take advantage of this. In my experiments I tried two different compression schemes. Since one outperformed the other (sometimes marginally, sometimes by as much as a factor 2) I will only describe the more successful; this, by the way, is not to say that it cannot be improved upon.

Having the sum of a 2x2 pixel square (pixels 1, 2, 3, and 4) available, the information transmitted to reconstruct the square is:

$$1 + 2 - 3 - 4$$
$$1 - 2 + 3 - 4$$
$$1 - 2 - 3 + 4$$

or, horizontal, vertical, and diagonal pair differences. These differences will generally be smaller than individual pixel values. The differences are calculated over an area of 4x4 squares, the number of significant bits, N, in the absolute biggest of the 48 differences is calculated, and all differences are offset by 2**(N-1) (to obtain unsigned integers). The bit stream is then built up by first encoding N in five bits and then the 48 differences, each in N bits; if N=0 (implying that all 16 2x2 squares have a constant intensity), nothing more follows after the five zero bits indicating N=0.

Obviously, the savings to be gained by such techniques depend on the contents of the map that is being transmitted. Mostly empty maps will allow much more compression than completely filled ones. I believe, though, that there will always some compression to be achieved.

5. RESULTS

I experimented with two VM maps, using only the inner quarter of each. One was a map of NGC 6251, which is mostly empty, the other a map of the Crab Nebula which was almost completely filled. The following table summarizes the results. Sizes are given in number of 80 character records, assuming 7 useful bits per character. The "Crab H" maps were histogram-equalized.

COMPRESSION RESULTS

Map	Number of bits transmitted	Subsection size (pixels)	Original size (recs)	Transmission file size (recs)	Compression factor
N6251	8 bits	256x256	937	61	15.4
N6251	12 bits	256x256	1405	158	8.9
N6251	16 bits	256x256	1873	604	3.1
Crab	8 bits	512x512	3745	1657	2.3
Crab	12 bits	512x512	5618	2828	2.0
Crab	16 bits	512x512	7490	4141	1.8
Crab H	8 bits	512x512	3745	2105	1.8
Crab H	8 bits	256x256	937	601	1.6

The last column gives just the compression achieved by the bit compression described in section 4. Combining the compressions resulting from this and the histogram equalization, one can conclude that a compression a factor of a little over 3 can be achieved for the Crab map, and something like a factor 10 on the NGC 6251 map. Experience has shown that the size of the transmission file roughly triples for each doubling of the map size (i.e., transmitting the last map in the table to half resolution, at a size of 128x128, takes about

200 records). The encoding requires, of course, initial overhead spent on reading the map and constructing the pyramid. After that, on the VAX, it takes about 30 msec per output record; most of this time is spent in constructing the bit stream.

6. CONCLUSION

Going back to the canonical example given in the introduction, sending all 16 bits in the inner quarter would take about 22 hours. Transmitting only 8 bits (if so desired, histogram-equalized) would bring it down to 11 hours. Transmitting a smoothed map (4x4 smoothing) would require about 40 minutes, while progressive transmission with bit compression requires roughly half that for the same map, with the option of improving the resolution on subsections of it at appreciably less cost than in straight transmission. The total map would take approximately 3 hours. "Standard" 512x512 8-bit images take 10 to 15 minutes, with a 128x128 version available after 1 or 2 minutes. This all assumes 1200 baud and error checking.

We can conclude that we can profit considerably from these more sophisticated data transmission techniques, certainly if Crays around the country are going to crunch our numbers and we would like to be able to monitor their progress from our own facilities without having to invest in expensive data communication lines. Images of reasonable quality and usefulness can be transmitted essentially at interactive speeds with the option of extension to full detail in times of the order of one or two cups of coffee.