ENHANCEMENTS TO THE PROGRESSIVE TRANSMISSION METHOD

S. E. Walker, Jr., University of Maine, Orono, Maine 04469, USA
F. S. Hill, Jr., University of Massachusetts, Amherst, Massachusetts 01003, USA

ABSTRACT

There are many instances where remote users might want to interactively access an image system and browse through files of large images, such as satellite or medical photos, to determine which ones to examine in detail. One could also envision the need for a "quick look" capability in a dynamic environment such as the cockpit of an aircraft, where a gestalt of a terrain/navigation map provides the pilot with immediate relational information, with the detail following later. Extensions and enhancements to the Progressive Transmission Method are described that permit efficient and lossless encoding and transmission of rich full color images, to support just such interactions with large image databases. The method is viewed in a "transmission cone" context which is an architectural extension of the familiar image pyramid structure. The transmission cone assists the user in interacting with large color images in a variety of system implementations. One particular implementation discussed is an "Interactive Image Query System."

KEYWORDS: Image analysis, progressive transmission, progressive refinement

I. Introduction

Rapid advances in VLSI and microprocessor technology are making powerful image analysis terminals available at low cost. We wish to provide interactive image query service to users working at display terminals at sights which are remote from an image database to allow them to receive and process images on a timely basis. A user may wish to browse rapidly through a 'slide tray' of images, or roam over a very large image. Often the user will need only a "quick look" through a slide tray to determine which images to view in full detail. One such situation is depicted in Figure 1 where the user accesses an Interactive Image Query System for remote-sensing images. In this paper we address the problem of interactively accessing images, and we relate this to an associated problem of retrieving computer generated maps in aircraft.

For an interactive system to be effective the user must be provided with enough data to spot key information quickly. Unfortunately, it requires a substantial amount of time for an image to be transmitted over a low bandwidth data link such as a telephone line. For a raster image with r rows, c columns, and b bits/pixel a total of r* c* b bits must be transmitted. For simplicity, we will define the 'typical image' as having 512 rows, 512 columns, and 24 bits per pixel, giving r.c.b. = 3/4 M Byte or 6.3 Mbits. We also define as 'standard' a transmission rate of 4800 bits/second, resulting in a transmission time of 21.85 minutes for the typical image. Since most systems transmit an image line-by-line, top-to-bottom, a user typically cannot determine whether an image is of interest until the screen is at least half full, which may take 10 minutes. Such slow acquisition of an image is frustrating to interactive users and reduces their productivity.

The problem has been addressed through the use of quad-trees and progressive refinement of images [Sloan and Tanimoto, 79] and extensions to binary trees using the Progressive Transmission (hereafter PT) Method [Knowlton, 80]. The PT method is a clever scheme to encode, transmit and decode an image so that the user receives pertinent information concerning the image early in the reception phase. The method is simple, efficient, and there is no loss of information if the decoding operation is allowed to continue to completion. We briefly describe the PT method to set a framework and nomenclature for discussion of the enhancements to it. For a detailed discussion of the PT method see Knowlton [80].

II. Overview of the Progressive Transmission Method

When using the PT method to transmit an image, the original r* c* b bits of the image would first be encoded into a new set of r* c* b bits at the archive site, and stored in encoded format. When an image is requested by a remote user, bits are sent so that the received image fills in as shown in Figure 2; the image is successively bisected (alternately in the horizontal and vertical directions) by sending b bit quantities which Knowlton called DIFFERENTIATORS (hereafter "diff"). Each diff combines with a displayed COMPOSITE ("comp") value to produce two new comp's for the new smaller pixels. The process begins by sending b bits for the comp of the single fat pixel, and each splitting requires b bits more. Hence, after M*b bits have been received the displayed image consists of M' fat' pixels.

Corresponding to the decoding process which goes on at the user's terminal, there is an encoding process which maps original pixel pairs into (comp, diff) pairs. This goes on at the image archive site, whereupon the image is stored in PT format. Adjacent pixels in the image are combined in pairs, the two comp values producing a single fat pixel comp and a new diff. The b bit color or intensity in each fat pixel approximates the average of all the small pixels contained within the fat pixel.

An important advantage in Knowlton's PT method is that all quantities are b bit integers, and the usual numerical roundoff error which would accumulate in finding exact averages is avoided. Instead of calculating averages in the encoding process look-up tables are used to map each pair of comp's into a comp and diff pair. It is convenient to view the encoding look-up table as a mapping $T_e(.,.)$ from b-bit value pairs into similar pairs: T_e(comp1, comp2) = (comp, diff). Figure 3a shows an example table used for encoding. A pair of comp values (e.g. (comp1, comp2) = (1,3)) are used as indices (e.g. row=1, col=3) and the table yields (comp,diff) = (2,0). A similar table and mapping $T_d(.,.)$ exists for decoding.

The encoding tables are arranged so that each comp will split into two new comp's whose average is as close as possible to the original comp. Inspection of the table in Figure 3a yields an error (error = true average - original comp) distribution as shown in Figure 3b. For instance, along the equal index diagonal the error is 0. In general, error values are bounded by 2 ** (b-2). Recall, however, that these errors do not accumulate at successive stages of the PT method. Since the encoding and decoding processes are inverses of each other, information is rearranged but not destroyed at each stage, and if the decoding is allowed to proceed to its conclusion there is no error in the recovered pixel values.

A significant limitation in the original PT scheme is the requirement for the two look-up tables. In Knowlton's examples each pixel was represented by only four bits, hence the tables were only 16 by 16. However, the much richer images associated with remotely sensed data may contain 24 bits/pixel, requiring a user to construct two 8.5 gigabyte tables to use PT: clearly an unmanageable task.

Presented below is an extension of Knowlton's method to allow its application to the very demanding area of rich, full-color images with any number of bits/pixel. The look--up tables are replaced with a SINGLE simple procedure that works for BOTH encoding and decoding. Its space complexity is independent of the number of bits per pixel.

III. Enhancements to PT Method

IIIa. Elimination of the Look-up Tables

We focus on the nature of the transformation T_e & T_d as in above which map ordered pairs of b-bit numbers into other ordered pairs. Knowlton reported one such T for 4-bit values, in which he distributed values in a clever but ad hoc way to build tables having small errors as in Figure 3b.

Since the table was constructed in this way, it is not clear whether a simple and compact algorithm exists that COMPUTES T_d (comp, diff) rather than looks up its values. We also need ways to extend the method to higher values of b. We first develop a simple geometric view of T which allows one to build a look-up table for any b. The tables agree exactly with Knowlton's reported cases. Using this structure we then construct a simple algorithm which replaces the table, at a tremendous space saving.

IIIb. The Ring Neighbor Algorithm

An image using b bits/pixel requires the equivalent of an M by M table, where M = 2 ** b. Call by N the set of possible pixel values: N = (0,1,...,M-1). Elements of the tables are ordered pairs with components taken from set N. Then the table may be viewed as a mapping from the Cartesian product N x N onto itself: the pair of indices (i,j) maps into the pair T(i,j). The table must be one-to-one and onto for an inverse transformation to exist; otherwise encoding and decoding would not be inverse operations, and PT would not be information lossless.

Figure 4 shows an example table template for b = 3 which reveals the structure of PT tables, and leads to a general algorithm. View the table as consisting of concentric rings. The innermost ring is composed of the central four elements, while the r-th ring has 4 * (2r-1) elements. We find the 'ring neighbor' of any cell in the r-th ring by traversing the ring r elements in a cw direction, as shown in Figure 4 for various rings. To build the table find the ring neighbor for each cell in the table, and place the INDICES of this neighbor in the cell. That is, for cell at indices i,j:

a. Determine which ring this cell lies with. Call it r.

b. Find the indices of its ring neighbor.

c. Interchange the indices.

d. Write them into cell (i,j) as the required ordered pair.

For example, in Figure 3a, cell (2,3) lies in ring 2, and has as a ring neighbor the cell at (0,3), so place (3,0) in the cell at (2,3). Note that a table built using this method will be identical to Knowlton's, except for a reflection about the i = j axis (a transposition), an operation which as no effect on the average error distribution of the PT method.

The implementation of this transformation as a procedure may be found in (Hill 83). It operates on integers and has a trivial space complexity independent of b. Each encode or decode operation requires at most 4 integer compares and 7 integer adds, (3 compares and 4 adds on the average).

A remarkable property of the tables which are generated by this approach is that they are INVOLUTIONS: i.e., T is its own inverse. applying the mapping twice recovers the original index pair: T(T(i,j)) = (i,j). This means that the SAME T, hence the same procedure, can be used for both encoding and decoding.

IIIc. Transmission Cone

It is very useful to view PT in the context of a 'transmission cone' as shown in Figure 5, analogous to 'processing cones' and 'quadtree pyramids' used in image processing [Hans 81, Pavl 82]. The encoding process at the archive site progresses down the cone until the single fat pixel at level 0 is reached, while decoding using a stream of diff's progresses up the cone to the level desired.

To obtain a level L + 1 image from one at level L requires the transmission of an additional 3 * (4**L) diff's taking four times as long to receive the data for each successive level. For instance, a user can receive a level 6 image in 1/64 the time needed for a level 9 image! Thus, it is very advantageous for a user to be able to decide at a low level whether or not an image is interesting.

Although the transmission cone is similar to processing cones and quadtrees, several significant points are worth noting:

a. The ring neighbor encoding algorithm replaces numerical averaging, uses only integer arithmetic and is information lossless.

b. Two passes over the image are required to move from one level to the next. The user can display the intermediate stages if desired to attain a smoother increase in resolution for a given volume of data.

c. In order to save memory at the user's terminal, only one level of the image is stored.

d. Like the process cone the transmission cone is envisioned as an array architecture that will allow its application to new and diverse image analysis problems. We envision the implementation of the transmission cone as having a "perfect shuffle" [Ston, T3] interconnection between array elements, a departure from most image processing architectures.

IV. Results

An interactive Image Query System for remote sensing images has been designed using these ideas [Hill 83]. A brief summary of some of its user functions is presented below.

a. The user "queries" the system for a set for a geographical region, and browses through them using simple commands. Each image is compressed to fit on the screen. For example, a 2048 x 2048 image would be inserted in the browse file at the level of the transmission cone that most closely matches the resolution of the user's screen; level 9 for a 512 x 512 screen.

b. The user gets a quick look at a low resolution version of the image in a fraction of the time normally required. As described earlier, the typical level 9 image of Figure 6 would take over 21 minutes to be received and displayed, while Figure 7 shows the same image at level 6 in the transmission cone requiring 20.48 seconds. Here the user has specified a portion of the low resolution image of particular interest, and relevant diffs have been sent to fill in this area to full resolution very quickly.

c. The user may examine portions of a very large image (such as a Goes Image, for which r = 1800, c = 3822, b = 8 [Rice, 80]). The user specifies a window of interest and the relevent diff's are sent from the appropriate subcone which represents the windowed region.

V. Ongoing Work

Work is continuing to refine and further develop the Interactive Image Query System as well as to move toward an architectural implementation of the transmission cone.

We are also pursuing several areas of research that offer promising results using the enhanced PT method and the transmission cone construct.

Va. Computer Generated Maps

Advances in computer generated map displays are making it possible to provide electronic map displays in highly dynamic environments such as aircraft cockpits. The map usually consists of some major 'Instrument flight rule' navigation fixes along with a rather austere and bland background color, having very little topographic information. We are working to present more meaningful displays to assist the pilot in rapidly grasping the full impact of the terrain.

The problem has similarities with that of the remotely situated image user, in that there is a data transfer bottleneck from mass storage to display. The response times demanded of the system are much shorter, negating the effect of the greater data transfer rates available. Our approach is to use an overlay showing rivers, coastlines, text, etc., generated from vastly compacted on-board datasets [Hill 82]. The background is added from pre-processed satellite or topographical information stored on-board in PT format. When the aircraft enters a new area, or when the pilot wishes to 'browse' through a set of maps this background is rapidly displayed at, say, level 6, giving the pilot the necessary 'quick look'. As the aircraft moves through the map area detail fills in gracefully by transferring additional data (moving up the cone), until full resolution is attained.

Vb. Human Factors Issues

There are many human factors issues that must be resolved to use the PT method effectively:

a. At what level of the transmission cone is meaningful information provided? Preliminary work indicates level 6. No definitive conclusion has been reached.

b. Must all 24 bits of a color image be encoded? During the browsing phase, is it possible that only 3-4 bits per color is needed?

c. What effect does color have? What is the effect on efficiency of the system and the user if a coarse image such as level 6 is sent with color suppressed?

VI. Conclusion

Knowlton's innovative scheme for transmitting images has been extended in several directions, with a particular eye to providing 'real-time' displays in which a remote user can interact with detailed color images such as remote-sensing images. The result has been the implemention of an elementary "Interactive Image Query System." Some limitations of earlier work have been surpassed and a variety of new approaches have been developed to provide the user with more flexibility in receiving the most interesting portions of images.

Acknowledgement

This work was supported in part by the Remote Sensing Center at the University of Massachusetts through the ASTECH Corp. The Visions group at the University graciously denoted the use of their facilities.

References

(Hans 81) A. Hanson & E. Riseman, "Segmentation of Natural Scenes," in COMPUTER VISION SYSTEMS (A. Hanson & E. Riseman, etc.), Academic Press. New York, 1978.

(Hill 82) F. S. Hill and S. E. Walker, Rapid Display of Compacted Map Data Sets on Board Aircraft," 4th Symposium on Automation in Engineering Data Handling, Conference Proceedings, Monterey, CA., November 1982.

(Hill 83) F. S. Hill and S. E. Walker "Interactive Image Query System Using Progressive Transmission", to appear in July 1983 SIGGRAPH 83. Conference Proceedings.

(Knowl 80) Ken Knowlton, "Progressive Transmission of Grey-Scale and Binary Pictures by Simple, Efficient, and Lossless Encoding Schemes," Proc. of IEEE, Vol. 68, No. 7, 1980, pp. 885-896.

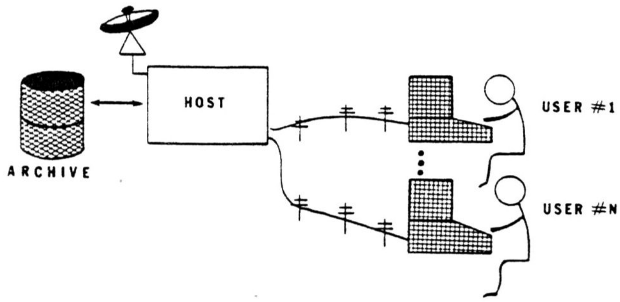
(Pavl 82) Theo Pavlidis, GRAPHICS AND IMAGE PROCESSING, Computer Science Press, Rockville, MD 1982.

(Rice 80) R. F. Rice & A. P. Schlutsmeyer, "Data Compression for NOAA Weather Satellite Systems," SPIE Vol. 249. ADVANCES IN IMAGE TRANSMISSION II, 1980, pp. 131-137.

(Sloa 79) K. Sloan & S. Tanimoto, "Progressive Refinement of Raster Images," IEEE Transactions on Computers, Vol. C-28, No. 11, November 1979.

(Walk 82) S. E. Walker, & F. S. Hill, "Progressive Transmission of Remotely Sensed Images," 4th Symposium on Automation in Engineering Data Handling, Naval Postgraduate School, Monterey, CA, November 3, 1982.

(Ston 71) Stone, H. S., "Parallel processing with the perfect shuffle", IEEE Trans on Computers, C-20, 153-61, 1971.



Interactive Image Query System

Figure 1. Configuration of Image Query System

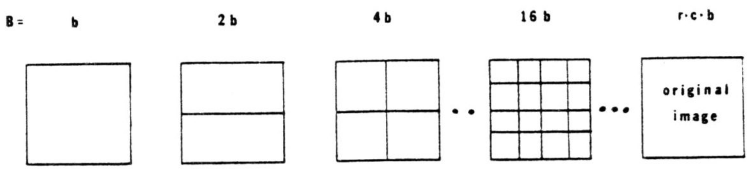


Image Seen After Receipt of B bits

Figure 2. Emergence of image as User Watches.

a.)

3	23	33	32	31
2	13	22	21	30
1	03	12	11	20
0	02	01	00	10
	0	1	2	3

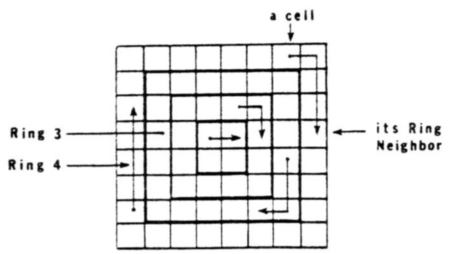
Look-up Table

b.)

3	$-\frac{1}{2}$	-1	$-\frac{1}{2}$	0
2	0	$\frac{1}{2}$	0	$\frac{1}{2}$
1	$\frac{1}{2}$	0	$\frac{1}{2}$	0
0	0	$\frac{1}{2}$	1	$\frac{1}{2}$
	0	1	2	3

Error Distribution

Fig. 3 Example Look-up Table and Error Distribution



Defining Ring Neighbors

Fig. 4 Table Template to Describe Geometry of Rings.

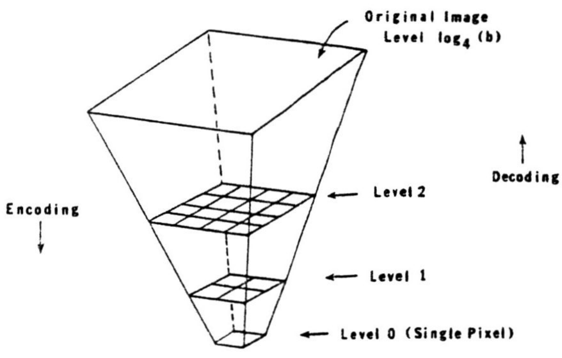


Figure 5. The Transmission Cone.

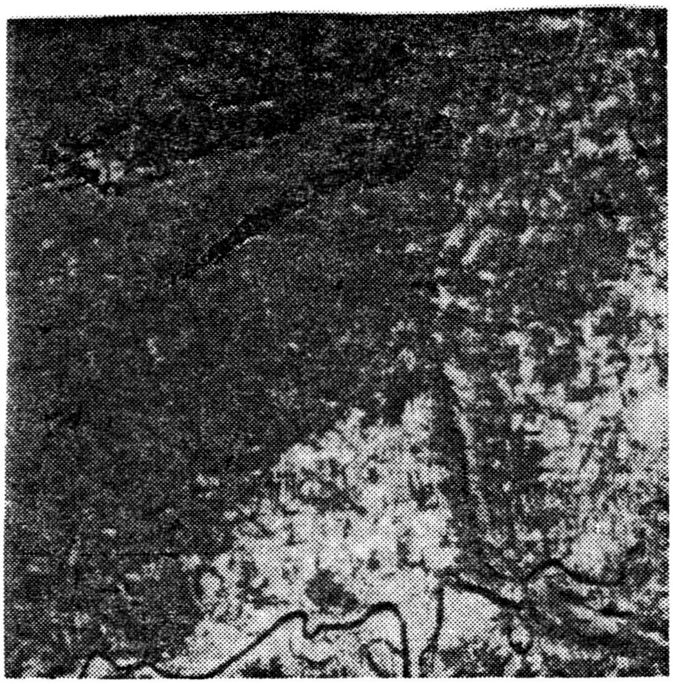


Figure 6. Example level 9 Image.

Figure 7. Level 6 Image with Level 9 Fill-in.