COMMUNICATIONS RESEARCH CENTRE
CENTRE DE RECHERCHE SUR LES COMMUNICATIONS

Trisignal Data Compression Optimization Study

Literature and Patent Search Report

Dr. Michael Sablatash
Communications Research Centre
3701 Carling Avenue
Ottawa, Canada K2H 8S2
Telephone: (613) 998-2061
Fax: (613) 990-0316

October, 1994

COMMUNICATIONS
RESEARCH CENTRE

CENTRE DE RECHERCHE
SUR LES COMMUNICATIONS

Trisignal Data Compression Optimization Study

Literature and Patent Search Report

Dr. Michael Sablatash
Communications Research Centre
3701 Carling Avenue
Ottawa, Canada K2H 8S2
Telephone: (613) 998-2061
Fax: (613) 990-0316

October, 1994

1.0 Introduction and Methodology

This interim report is submitted after 21 days of effort, only part of which will be charged to Trisignal Communications. Within the limits of the time and financial constraints, a preliminary literature search was conducted. A search was conducted for key words including, "Lempel-Ziv, data compaction, lossless data compression, adaptive and lossless compression or coding, adaptive and data compaction, adaptive and universal coding, adaptive compression, Lempel-Ziv or Ziv-Lempel, LZW,and LZS". These keywords and combinations of them were used to search on-line computer databases. The databases and the time periods searched were: INSPEC, 1969-1994; NTIS, 1964-1994; Engineering Index Compendex*Plus(TM), 1970-1994; Pascal, 1973-1994. The following numbers of items were found from various combinations of keywords.

(1)Adaptive and (lossless(3n) (compression or coding)), 68 items;

(2)Adaptive and data(w)compaction	2 items;
(3)Adaptive and universal coding	3 items;
(4)Adaptive(5n)compresseon	742 items;
(5)Set(4) and lossless	40 items;
(6)Lempel(n)Ziv? or Lempel-Ziv or Ziv-Lempel	260 items;
(7)LZW	67 items;
(8)LZS	22 items.

Various other other combinations were tried also. With the help of Jean Belleau the items obtained from the INSPEC database were prioritized into three classes, 1,2 and3, with class 1 being the highest priority. Twelve of the items were in class 1, 11 in class 2, and 8 in class 3. The author also searched literature that was too new to be included in the computer databases. The author studied material in 4 books to 'get up to speed', 10 articles in class 1, 7 articles in class 2, eight articles in class 3 and several others. About 25 articles were studied less thoroughly. These are all contained in the annotated bibliography of 50 items in Section 2.0. A substantial number of items have still not arrived at the library although they were almost all requested before Oct. 1.

An on-line search of the U.S. and Canadian patent literature was also conducted. Thirty-four items were retrieved, and eight judged relevant. These are listed in Section 3.0.

2.0 Reference List with Annotations

1. R. W. Lucky , Silicon Dreams: Information, Man, and Machine.New York: St. Martin's Press,1989, Chaps. 2&3, pp. 37-145.

There is a very insightful and physical treatment of essential concepts of information theory needed for text processing and compaction in Chapter2. Almost no mathematical formulae appear. In Chapter 3 a holistic treatment is given of information and the evolution of written language; language, information, and computers; the layers of text (in "onion-skin" layers from the outer layer: letters,words, syntax, semantics,and meaning); the statistics of English letters; text compression;and a quest of entropy on a deeper level. Very helpful tables are provided showing sizes of typical documents in computer storage units, transmission times for typical documents, frequencies of occurrence of individual letters in English text, entropies of approximations to English text(in bits per letter), portion of Lempel-Ziv dictionary for Chap. 2 of the book, Lempel-Ziv compression for Chap. 2, frequency of most frequently used words, and entropy of English in bits per letter, are given in Chap. 3. Lempel-Ziv coding is very insightfully explained in Chap. 3. This is an important first work to read for fundamental concepts and insights. In conclusion, this book represents an excellent starting point. It is comparatively easy to read, and in some places it is even entertaining!

2. N. Abramson, Information and Coding. New York: McGraw-Hill,1963, Chaps.1-4, pp. 1-92.

This is an important fundamental work to obtain the necessary foundations in information theory and the basic mathematical foundations for statistical source coding. It includes the clearest explanation of Huffman coding that I know. The mathematics is not demanding, and quite a number of very helpful examples are provided. This is an essential first introduction to the analytical and statistical aspects of source coding.

3. R.E.Blahut, Digital Transmission of Information. Don Mills, Ont.:Addison-Wesley Publishing Co.,1990, Chap.7, Data Compaction Codes, pp. 292-322.

This book includes sections on discrete sources, prefix codes for data compaction, tree codes for data compaction, and universal codes for data compaction. The Huffman, Elias, Ziv-Lempel, and the Miller-Wegman variation of the Ziv-Lempel code are described, with examples. The mathematical level is not too demanding. This is a relatively short introduction to both the methods that require a probabilistic model of the source ,such as the Huffman and Elias codes, and universal codes, such as the Ziv-Lempel family of codes, which can be used when probabilistic models are not known *a priori*, and a model must be constructed from the data set itself. I found this a useful and clearly written chapter which serves well as a transition from the previous ones to the modern and the more mathematical treatments in the journal literature, and in the more advanced and specialized books.

4. T.M.Cover and J.A.Thomas, Elements of Information Theory. Toronto: John Wiley &Sons, 1991, Chap.5, Data Compression,pp.78-124, Chap.6, Gambling and Data Compression,pp.125-143, and Chap.12, Information Theory and Statistics, pp.279-335.

This is a modern and quite rigorous and sophisticated treatment of the subject of data compaction (not lossy compression). The mathematical level is mature, but it is in line with that required to understand the analytical journal articles and more advanced books, so it can be regarded as the final stepping stone before really digging into the subject. Illustrative examples are included which are quite helpful. In Chap. 5 preliminaries such as the Kraft inequality,bounds on the optimal codelength, and the Kraft inequality for uniquely decodable codes are covered first. Then Huffman codes and their optimality are presented, in a thorough manner. The Shannon-Fano-Elias coding is presented next, and then arithmetic coding. Then the competitive optimality of the Shannon code is treated. It should be obvious that the codes described are for sources for which the probabilities are known a priori.

Chap. 6 provides a refreshing connection between gambling and data compression. Section 6.4 discusses the entropy of English. Section 6.5 makes the connection between data compression and gambling, and the last section, Section 6.6, describes the gambling estimate of the entropy of English.

Chap. 12 describes the method of types in the first section, and discusses various other statistical theory and its relation to information theory before providing a rigorous and sophisticated treatment of Lempel-Ziv coding.

This is necessary reading to be in tune with the knowledge many graduate students are acquiring, and to be able to read the advanced articles in the area. While it is strongly recommended for anyone planning to undertake research and development in this area, it is probably at a greater theoretical depth than is necessary for applying techniques invented and described by others.

5. L.D.Davisson, "Universal Noiseless Coding", IEEE Trans. on Inf. Theory,vol.IT-19, pp.783-795, Nov. 1973.

This is a fundamental paper defining universal coding. A sufficient condition for the existence of universal codes is given and proven. In the section on histogram encoding code constructions of great generality are given for use when very little is known of the source. Fixed-rate coding employed blockwise rather than variable-length coding is described.

This is an important fundamental paper for serious work on data compaction.

6. A. Lempel and J.Ziv, "On the Complexity of Finite Sequences", IEEE Trans. on Inf. Theory, vol.IT-22, pp.75-81, Jan.1976.

This is a paper providing deep and fundamental insights into the complexity of finite sequences. This enables an evaluation of the complexity of the sequences obtained in source coding This paper is often cited in analytical studies of data compaction algorithms.

7. J. Ziv and A.Lempel, "A Universal Algorithm for Sequential Data Compression", IEEE Trans. on Information Theory, vol. IT-23, pp. 337-343, May 1977.

This is one of the two famous papers which have led to practical algorithms for data compaction. This one is now often referred to as LZ77. The methods derived from LZ77 are computationally intensive for encoding, but decoding is very efficient. Thus, LZ77 is a good method when a file is to be decoded many times, or is to be decoded on a smaller machine. Much work has been done to increase the speed and decrease the complexity of the encoding of LZ77.

8. J. Ziv and A.Lempel, "Compression of Individual Sequences via Variable-Rate Coding", IEEE Trans. on Information Theory, vol. IT-24, pp.530-536, Sept.1978.

This is the other one of the two famous papers which have led to practical algorithms for data compaction. This one is referred to as LZ78. The methods derived from LZ78 strike a different balance from those derived from LZ77, with both encoding and decoding requiring a moderate amount of resources. Thus, methods derived from LZ78 are more appropriate for files that are not expected to be decoded often (forexample, archives, backups and electronic mail).

Both this reference and the previous one are important to understand if fundamental advances are being pursued, and if the variations of LZ77 and LZ78 are to be appreciated.

9. M.R.Nelson, "LZW Data Compression", Dr. Dobb's Journal, pp.29-87, Oct.1989.

The acronym stands for Lempel-Ziv-Welch. This article familiarizes the reader with data compression programs such as ARC, PKZIP, COMPRESS, and COMPACT. LZW is described without mathematics, but the algorithms and processes for compression and decompression are given. Implementation problems and solutions are discussed. There is a short discussion of portability. This is a useful paper to read for an introduction to the practical implementation and problems of data compaction. It provides a balance to the study of more theoretical and deeper papers. It requires some experience to fully appreciate this paper, as well as a knowledge of C.

10. H.C.Kotze and G.J.Kuhn, "An Evaluation of the Lempel-Ziv-Welch Data Compression Algorithm", COMSIG 1989, Proc. of the South African Conference on Communications and Signal Processing,pp.65-69.

In this paper the Lempel-Ziv-Welch algorithm is evaluated for the removal of the redundancy in computer files. The LZ algorithm, and related algorithms, are compared to the LZW algorithm using encoding and decoding speed, memory requirements, and the compression ratio as performance criteria. Brief descriptions of the types of redundancies--character distribution, character repetition, high-usage patterns and format redundancy--are given. Summary non-mathematical descriptions of Huffman coding, run-length coding and the LZ algorithm are presented, the latter under the heading "Adaptive compression techniques". The LZW technique is described in more detail, including the exact steps in the encoding and decoding algorithms, and an example of each of these algorithms is provided. Data structures for encoding and decoding algorithms to decrease memory for encoding and decoding, and to increase speed. Results are shown in graphical form for compression ratios for larger files, and other types of files, and for coding time and memory requirements. One conclusion is that it is feasible to implement the LZW algorithm software.

This paper is short on theory and explanations, but it provides some bases for using the LZW algorithm to compress computer files.

11. V.S. Miller and M.N.Wegman, "Variations on a Theme by Ziv and Lempel", Proc. IEEE International Conference on Communications '88, vol. 1, pp.390-394.

This is an important paper which was first presented at the conference "Combinatorial Algorithms on Words" in 1985, whose proceedings were published by Springer-Verlag in 1987. The LZ algorithm will encode long strings of source symbols into code at close to the source entropy but will then append the innovation symbol in uncoded form. This means that while all other symbols encode into one or two bits, the innovation symbol requires eight bits. The inefficiency due to a noncompacted innovation symbol may be negligible if the encoded strings are long enough, but in practice the strings are of moderate length and the inefficiency is not negligible. One of the improvements to the LZ algorithm described in this paper is to eliminate this inefficiency of using an uncoded innovation symbol by leaving the specification of the innovation symbol pending until after the next iteration. New strings are not fully defined until after a delay of one iteration. The innovation symbol is retained as the first symbol of the data waiting to be encoded. After decoding the next codeword, the innovation symbol can be recovered and appended as the last symbol of the previous string. Two more improvements are the ability to use fixed size encoding tables by using a replacement strategy, and more rapid adaptation by widening the class of strings which may be added to the dictionary. These improvements also provide an adaptive probabilistic model for the input data. The issue of data structures for efficient implementation is also addressed. Extension and possible future work are discussed.

This paper requires a substantial background in data compaction to understand. It contains no mathematical equations, but assumes a sophisticated reader. The original LZ algorithm and algorithms for the improvements are given.

12. P.E. Bender and J.K. Wolf, "An Improved Sliding Window Data Compression Algorithm Based on the Lempel-Ziv Data Compression Algorithm", Proc. GLOBECOM '90: IEEE Global Telecommunications Conference and Exhibition--Communications: Connecting the Future, San Diego, CA,Dec.2-5,1990, pp.1773-1777.

This paper presents very clear explanations accompanied by very helpful diagrams with windows illustrating encoding and decoding by the LZ and improved LZ algorithms. Indeed, the explanations are among the clearest to be found in the literature. Graphs show the comparisons among the LZ, LZW and improved LZW compression ratios for different file types. The improvements over the compression ratios of the LZ and the LZW algorithms are very impressive indeed for all three file types tested. I would definitely recommend implementation and extensive testing of the improved algorithm. This paper and the more extensive journal paper based on it (see item 13, below) are very important to read and consider for practical use.

13. P.E. Bender and J.K. Wolf, "New Asymptotic Bounds and Improvements on the Lempel-Ziv Data Compression Algorithm", IEEE Trans. on Information Theory, vol.37, pp. 721-729, May 1991.

This paper provides extremely clear explanations, along with the extraordinarily instructive diagrams with sliding windows used in the preceding reference, for the LZ and improved LZ algorithms. In addition analytical expressions for the compression ratios are derived. A theorem is proven which shows that in the limit as the sliding window size approaches infinity, the differential length of the matched string relative to the length of the matched string is, with probability 1, equal to zero. A brief overview of the LZW algorithm is given. A software implementation is described. The same graphs comparing compression ratios among the LZ, LZW and improved LZ algorithms are given as in the immediately preceding paper. As noted there the compression ratios for three different files are considerably greater for the improved LZ algorithm. This algorithm has an encoder of the same complexity as the original algorithm's encoder, and a decoder of the same complexity as the encoder Again, this algorithm should be studied for practical use. It will likely do better and can never do worse than the original algorithm.

14. C. Rogers and C.D. Thomborson, "Enhancements to Ziv-Lempel Data Compression", Proc. of the 13th Annual International Computer Software and Applications Conference, Orlando, FL., Sept. 20-22, 1989, pp. 324 -330.

This paper is full of excellent comparisons among commercial, LZ and four modifications of LZW algorithms, using six different test text files. A description of the LZW algorithm is given, with illustrative examples in tabular form. Huffman and arithmetic coding are briefly described. Then the ZLWAx family of coders are described. They build their dictionaries using the Ziv-Lempel-Welch algorithm and represent dictionary entries using arithmetic coding. Members of this family employ a predictive strategy that takes advantage of the arithmetic coder's ability to assign different probabilities to different dictionary entries, and two enhancements to the basic LZW algorithm. The predictive strategy(ZLWAP) tries to exploit the idea of context in the source text. A source text can contain any number of of contexts. String extension(ZLWAS) adds to the dictionary the concatenation of the last two dictionary entries for which codes were emitted. String extension makes the dictionary grow faster and contain longer strings, so many sources can be represented by fewer dictionary entries. This idea is used in the ZLWAS coder to allow it to adapt quickly to highly repetitive text when long dictionary entries have a good chance of being used. Word based strategy(ZLWASC) is a conditional dictionary entry strategy that often reduces the number of useless character extensions for a ZLW coder with the string extension enhancement. The idea behind conditional dictionary entry is that when the entries in a dictionary are letters or parts of words, it makes sense to character extend them until they become words. However, once you have a dictionary of words, it makes little sense to build long phrases character by character. It makes more sense to build them up by putting words together. The last of the family adds prediction to give (ZLWASPC). Two ZL (or LZ) type coders are included among those compared. They are UNIX Compress, a standard implementation of the ZL algorithm. One is ZL12, using a dictionary size of 4K, and the other ZL16, using a dictionary size of 64K. These ZL algorithms are compared with five others. These are:

(1) UNIX Pack,a standard implementation of Huffman coding(PACK);

(2) UNIX Compact, a standard implementation of adaptive Huffman coding(CPACT);

(3) 256-character move to front list using Huffman coding(MTF);

(4) Bentley-Sleator-Tarjan-Wei coding with list length239 and a maximum word length of 16 characters(BSTW); and

(5) Miller and Wegman's A2 algorithm(MWA2).

The compression ratios of the above 11 algorithms were compared on six text files. These were:

(1) a file of student essays;

(2) a Pascal source code file;

(3) the TEX source to the complete TEX manual file;

(4) a file of USENET news articles;

(5) a file of three repeated characters; and

(6) an executable binary program file.

Tables of dictionary statistics and run times are given. Future directions are discussed, and include improving the speed of the arithmetic coder, enhancing the conditional dictionary entry strategy or adding a first order Markov model to the predictive strategy. This paper provides many useful results and choices of encoders for implementation.

15. H. Yokoo, "Improved Variations Relating the Ziv-Lempel and Welch-Type Algorithms for Sequential Data Compression", IEEE Transactions on Information Theory, vol.38, pp.73-81, Jan.1992.

This paper is a gem! It is particularly noteworthy for its introduction of a powerful notation and analytical formulation of the LZ78, LZ77, and LZW algorithms and three new variations introduced in the paper, its analysis of their performance,and its analytical formulation of context gathering. The insight provided by this analysis of context gathering is applied to relate one of the new algorithms to Rissanen's Context algorithm. A diagram is shown relating the above seven algorithms. Two of the new algorithms are LZW-type practical algorithms derived from the third new one. The latter one is related to Rissanen's Context algorithm. The encoding and decoding are given as programs. Simulation results are given for five different files. One of the practical new algorithms is superior to the other one and to the LZW algorithm in compression ratio and in table size. This is a deep, unifying and insightful paper, which also provides practical information. The new algorithms are of interest for implementation, and the performance of one of these is particularly appealing. The paper is quite difficult because of its sophisticated, rigorous level.

Figure 1 shows relationships among compaction algorithms described and analyzed in this paper. It is meant to show that the LZW method was derived in 1984 from LZ78, that algorithm 2, derived by Yokoo, is intermediate between LZ77 (algorithm 3) and LZ78 (algorithm 1),that algorithm 4 is a practical version of algorithm 2, derived from it in a way similar to the way in which LZW was derived from LZ78, and that algorithm 2 and Context have opposite directions of context gathering, as explained by Yokoo.

16. G. Battail and M. Guazzo, "On the Adaptive Source Coding", Proc. EUROCODE '90: International Symposium on Coding Theory and Applications, edited by G. Cohen and P. Charpin, Springer-Verlag, Berlin, 1991, based on the conference in Udine, Italy, Nov. 5-9, 1990.

This paper is intriguing because it suggests a way of concatenating two compaction algorithms with the LZ algorithm to obtain improved performance by taking advantage of the complementary properties of the three algorithms. The two algorithms are the Huffman-Gallager and the Guazzo algorithms. A transition diagram in which the tree leaves merge with its root is used to describe the Huffman-Gallager algorithm. The LZ algorithm can be used to constructing a diagram of the same shape. When the probability of each branch conditioned on the node it departs from is constant and known, the Guazzo algorithm is a means for describing any path in such a diagram, hence any source

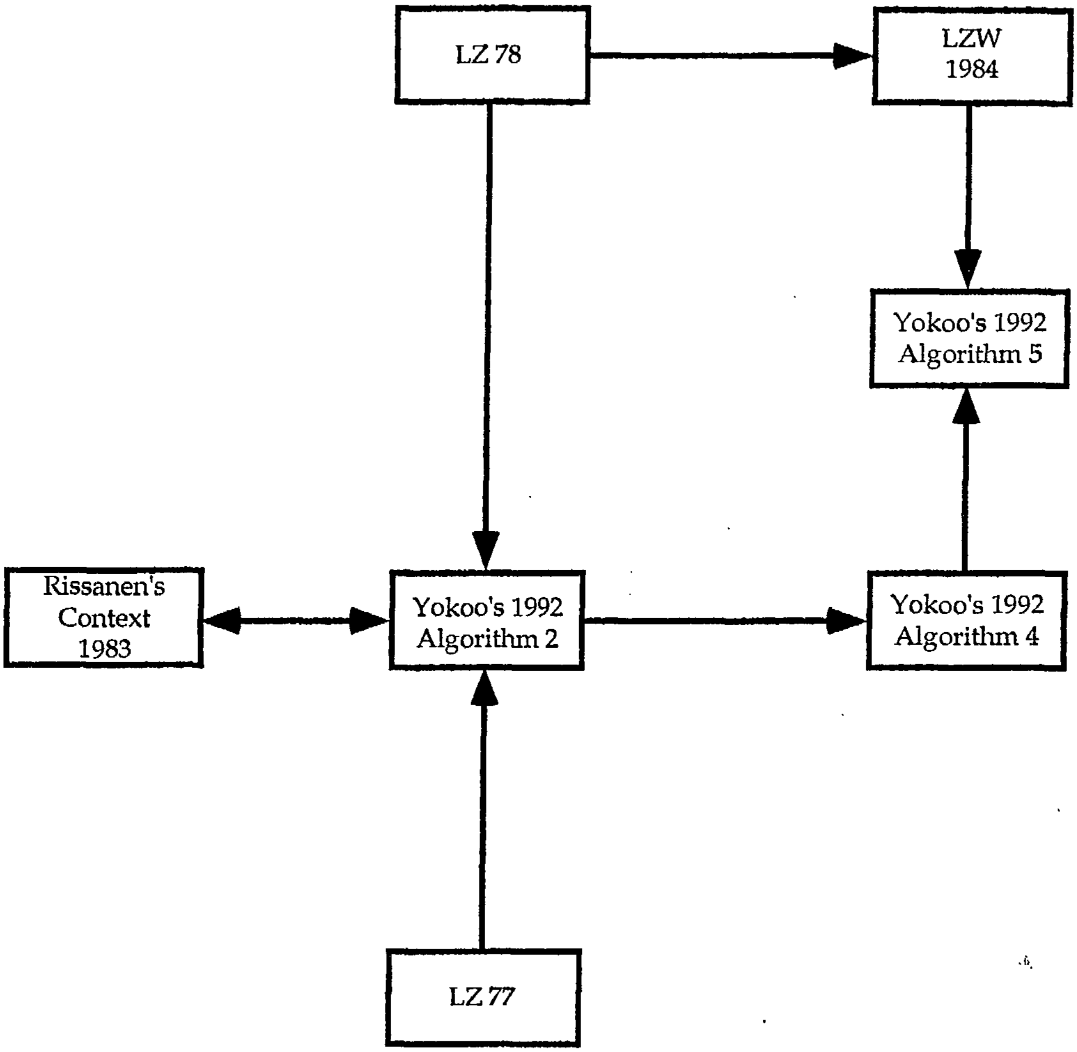

Figure 1: Relationships among algorithms analyzed and innovated by Yokoo (1992).

output. It results in a coded message whose average length approaches the theoretical limit as a closeness criterion is made arbitrarily tight. Its adaptive version replaces the conditional probabilities by frequencies. The paper describes how the source can be coded first by the LZ algorithm to provide a secondary source which best takes into account the source memory for a given accepted complexity. The Huffman-Guazzo algorithm then operates on this source, resulting in the best tree of shortest average length Finally, the adaptive Guazzo algorithm provides the average asymptotically shortest description of any path in the transition diagram corresponding to this tree, as its closeness criterion gets tight. There is only one mathematical equation and so no analytical results, and no performance results to enable evaluations and comparisons to be made. Therefore, the paper is interesting for its ideas and concepts, but lacks concrete results for evaluation.

17. K. Holtz and A. Lettieri, "Self-assembling Data Trees Yield Near Optimum Data Compression and Encryption", Northcon Conference Record, Portland, Oregon, Oct.1-3, 1991, pp.171-176.

This paper takes a novel approach to attempt to overcome the limitations in encoding speed of LZ type algorithms, which makes them unsatisfactory for very high speed operations such as for ISDN or LANS. Codebook encryption may yield nearly unbreakable codes but suffer the same speed limitations. Very high speed text compaction and encryption are claimed to be made possible in this paper by using 'autosophy' data trees which grow, from text input, like 'information crystals' in a memory without programming. AUTOSOPHY is described as a new science dealing with self-assembling structures such as crystals, living trees, societies or self-growing information networks. One figure shows text as a serial progression from bit by bit to character by character to word by word to sentence by sentence to paragraph by paragraph to section by section and, finally, a book to demonstrate how higher level symbols may be formed from lower level ones. Tree networks are qualitatively compared with LZ compaction. Algorithms are shown in block diagram form. Text compression by generating tree networks results in encryption because data retrieval is only possible for users in possession of the proper network tree copy. No equations are given, and no simulation results to provide comparisons. Thus, the paper provides ideas and techniques which must be further explored for practical application. The concepts are interesting.

18. T. Bell and D. Kulp, "Longest-match String Searching for Ziv-Lempel Compression",Software--Practice and Experience, vol. 23,pp.757-771, July 1993.

This is an important, practically useful paper. It explores techniques for speeding the encoding of the LZ77 family of data compaction algorithms, including LZSS, LZH and LZB. The common problem of the LZ77 family of compressors is to search the window for a longest match when N is large, and the paper concentrates on this problem. Eight searching algorithms are described and compared. Empirical evaluations are made, with the results displayed in graphical form. Each of the data structures represents a trade-off between average compression time, worst-case behaviour, and memory requirements. The performance tradeoffs among the data structures are discussed in the conclusions. This is an excellent paper, but not easy to read unless one is quite expert in this area. It is a very useful paper to study for selecting practical implementations. I highly recommend that it be studied.

19. M. Cohn, "Ziv-Lempel Compression with Deferred Innovation", Chap. 6 in Image and Text Compression, edited by James A. Storer,Kluwer Academic Publishers, Boston,1992, pp.145-157.

While Compress and other deferred-innovation compressors achieve respectable performance on highly compressible data(say 2-1 or better), their compression is disappointing on relatively incompressible data. In the extreme of total incompressibility, such as previously compressed or encrypted data, Compress frequently expands the input by about 45% when the output word size is 12 bits and by about 90% when the output word size is 16, to mention two common options. These stand in contrast to LZ realizations without deferred innovtion, where random data are expanded by about 5% for output words of 12 or more bits. The purpose of this paper is to explain the expansion caused by deferred innovation. Mathematical analysis is given. The paper is at an advanced level. It should provide insights for design of deferred-innovation compaction algorithms.

20. D.S. Hirschberg and D.A. Lelewer, "Context Modeling for Text Compression", Chap.5 in Image and Text Compression, edited by J.A.Storer,Kluwer Academic Publishers, Boston, 1992, pp. 113-144.

The LZ algorithms are regarded as competitors that must be reckoned with by those engaged in research on context modeling for text compaction. Yet context modeling has emerged as one of the most promising new approaches to compacting text. A finite-context model is a probabilistic model that uses the context in which input symbols occur (generally a few preceding characters) to determine the number of bits used to code these symbols. This paper provides an introduction to context modeling and recent research results that incorporate the concept of context modeling into practical data compaction algorithms. Early context-modeling algorithms are described and compared. A brief discussion of other competing methods states that at present the most commonly used data compaction systems are not context-modeling methods and that the UNIX utility Compress and other LZ algorithms employ dictionary models and fixed codes. Compress encodes and decodes using 450 Kbytes of internal memory and at a rate of about 15,000 characters per second (cps). A more recent LZ technique, algorithm FG, provides superior performance of 2.95 bits per character (bpc) over the corpus, compared to 3.64 bpc for the older method. It uses less memory (186 Kbytes for encoding and 130 Kbytes for decoding), but executes only about half as fast as Compress (encoding 6000 cps and decoding 11,000 cps). Experimental results are given for two new context model-based methods, and compared to UNIX Compress, FG and one of the best early context model-based methods, called PPMC. In the final section the future of context modeling is discussed. It is relatively new and very promising. Finite state modeling is an extension of finite-context modeling that permits exploitation of characteristics of the input that cannot be represented in finite-context models. Finite-state modeling is attractive but, to date, successful use of this increased potential has eluded researchers.

Context modeling has great promise, and this book chapter is one of the best works on the subject. It is the most promising alternative to LZ methods. Practical algorithms are described in the book chapter or referred to.

21.S. Hayashi, J. Kubo, T.Yamazato and I.Sasase, "A New Source Coding Method Based on LZW Adopting the Least Recently Used Deletion Heuristic", Proc. IEEE Pacific Rim Conference '93.

A new source coding method based on the LZW algorithm is presented. The method assigns variable-length codewords instead of LZW's fixed-length codewords. A least-recently-used deletion heuristic queueing buffer, each of whose entries has a different parsed string, is described. The proposed method yields better performance of the compression ratio than the LZW method without losing the benefits of the latter.The results of simulation runs are shown, for various files, and comparisons with the Huffman and LZW algorithms. This is a good paper, containing a new compaction algorithm, which could be useful in practice.

22. H.D.Jacobsen, "Some Measured Performance Bounds and Implementation Considerations for the Lempel-Ziv-Welch Data Compaction Algorithm",Proc. International Telemetering Conference, San Diego, Oct.26-29, 1992, vol.XXVIII, pp. 383-388.

This paper documents an implementation of the exact LZW algorithm. Experimental results are shown in graphical form for the compaction ratio of image and pre-compacted data, and for ASCII text as a function of dictionary size. This paper is helpful for understanding the performance limitations of the LZW algorithm. A simple table look-up implementation for the LZW algorithm is given in Basic.

23.G.S. Yovanof, "Recent Advances in Lossless Coding Techniques", Proc. International Telemetering Conference,Las Vegas, Nov.4-7, 1991, pp.7-18.

This is a good review. Sequential data compaction schemes are described first and then 2-D lossless coding schemes. Applications and hardware solutions are described. Simulation results are shown in tables. Seventy-seven references are cited.

24. C. Thomborson, "The V.42 bis Standard for Data-Compressing Modems", IEEE Micro, Oct. 1992, pp. 41-53.

This paper is an excellent description of the standard and its bases in LZ, LZW and the Miller-Wegman variation. It includes information about typical modems in which it is used.

25. T.C. Bell, J.G. Cleary and I.H.Witten, "Text Compression". Englewood Cliffs, N.J.: Prentice-Hall, 1990.

Probably the most comprehensive book on the subject. Includes the complete LZ and LZW families to the date of publication.

26. M. Nelson, "The Data Compression Book". San Mateo, CA: M&T Books, 1992

The book with the most algorithms of many kinds, and good explanations of how they work.

27. G. Held and T.R.Marshall, "Data Compression",3rd ed. Toronto: John Wiley & Sons, 1991.

An excellent introductory book.

28. J. A. Storer(ed.), "Image and Text Compression". Boston: Kluwer Academic Publishers, 1992.

Part 1 of this book, on image compression, contains 3 chapters, one on image compression, the second on fractal image compression using iterated transforms, and the last one on optical techniques for image compression.

Part 2 contains 4 chapters, an excellent first chapter on practical implementations of arithmetic coding, which I highly recommend,an excellent second chapter on context modeling for text compression, which I have discussed in more detail earlier, an advanced third chapter on ZL compressors with deferred innovation, and an excellent fourth chapter on massively parallel systolic algorithms for real-time dictionary-based text compression.

Part 3. the last part, is on coding theory. The first chapter in the section is on variations on a theme by Gallager, the second is on the coding delay of a general coder, and the third and last one is on finite state two-dimensional compressibility.

While some of the topics covered may not be directly relevant to Trisignal's line of business, there is much of both theoretical and practical value in this book. The 73-page bibliography on data compression is the most comprehensive I know of.

29. H. Yokoo, "A Lossless Coding Algorithm for the Compression of Numerical Data", The Transactions of the IEICE, vol. 73, pp. 638-643, May 1990.

This paper proposes a lossless coding methos for the compression of computer files of floating point or fixed precision numbers. The method is adaptive and requires no prior knowledge about the input data. Although it is quite simple and all that is needed is the incremental parsing technique by Ziv and Lempel, the proposed method compresses well any independent identically distributed (i. i.d.) sequence of numerical data generated by a source with a smooth distribution. In order to evaluate the performance and the convergence properties, a bitwise equivalent model is introduced, which combines the Ziv-Lempel type data compression methods with the probabilistic framework. The model shows that, for any sufficiently long i.i.d. sequence, the poropsed method attains the entropy with only a loss of 0.36 bits per word or so. Computer simulation results are also presented in support of this evaluation.

30. D.R. Helman, "General Purpose Data Compression ICs", COMPCON SPRING '91, Digest of papers, San Fransico,CA, Feb. 25-March 1, 1991, pp.344-348.

Three commercially available data compression ASICs are compared. All three are suitable for lossless compression of general data including text, object code, databases, etc. All use a form of Ziv-Lempel adaptive coding and can be integrated into a standard microprocessor system. Compression performance is compared and found to be similar, in the general vicinity of 2 to 1.

This is a useful article if implementation in state-of-the-art hardware is being considered.

31. A.D.Wyner, "Fixed Data Base Version of the Lempel-Ziv Data Compression Algorithm", IEEE Trans. on Information Theory , vol.37, pp. 878-880, May1991.

It is demonstrated that a variant of the Lempel-Ziv data compression algorithm,where the data base is held fixed and is reused to encode successive strings of incoming input symbols is optimal, provided that the source is stationary and satisfies certain conditions (e.g., a finite-order Markov source). An advanced, high-level paper, with rigorous analysis, but it will take some effort to evaluate the algorithm on real data, on typical computers, and to work out implementation details. Furthermore, the utility of the approach depends heavily on the assumption of stationarity, which would have to be validated for the traffic of interest to Trisignal.

32. G. Conn, "Real-time Compression Saves Memory", Electronic Product Design, pp. 22-23, 1993.

Describes the implementation, in hardware and in software, of the STAC compaction algorithm, and its advantages. Block diagrams are provided to show the device realizations in software and in silicon. A helpful short, practical paper for those who wish to use the STAC algorithm (which is based on the LZ algorithm), and obtain ideas for implementations.

33. J. Jiang and S.Jones, "Word-based Dynamic Algorithms for Data Compression", IEE Proc.-1, vol.139, pp. 582-586, Dec. 1992.

A new word-based dynamic LZ (WDLZW) algorithm for data compaction is introduced The novel feature is that the algorithm is optimized for the compression of natural language data, in which all the spaces between words are deleted whenever copy codes or literal codes are sent out. Therefore, better compression rates can be achieved. The algorithm can still compress alternative forms of data. The structure, operation and implementation of the WDLZW is described. A comparison with other algorithms when compressing a wide range of data forms is reported. For text-based information WDLZW offers attractive performance. For other forms of data, WDLZW provides compression rates similar to those of dynamic Lempel-Ziv systems.

This represents a new variation of the LZ algorithm that should be considered as a candidate for future applications. Algorithms in program form are given for encoding and decoding.

34. H. Morita and K.Kobayashi, "On Asymptotic Optimality of a Sliding Window Variation of Lempel-Ziv Codes", IEEE Trans. on Information Theory", vol. 39, pp. 1840-1846, Nov. 1993.

LZ77 parses input data into a sequence of phrases, each of which is the longest match in a fixed-sized sliding window which consists of the previously encoded M symbols. Each phrase is replaced by a pointer to denote the longest match in the window. Then, a window slides to just before the next symbol to be encoded, and so on. In this paper the LZ77 algorithm is modified to restrict pointers to starting only at the boundary of a previously parsed phrase in a window. Although the number of parsed of phrases should increase more than those in LZ77, the number of possible positions to be encoded is much smaller. Then, it is shown that for any stationary finite state source, the modified LZ77 code is asymptotically optimal with the convergence rate O(log log M / log M) where M is the size of a sliding window. This is a deep and rigorous paper at an advanced level, and includes sophisticated mathematical analysis. The new modification of the LZ77 algorithm should be evaluated for possible applications. Considerable effort is required for implementing and evaluating the algorithm.

35. P. Grassberger, "Estimating the Information Content of Symbol Sequences and Efficient Codes", IEEE Trans. on Information Theory, vol. 35, pp. 669-675,May 1989.

Several variants of an algorithm for estimating Shannon entropies of symbol sequences are presented. They are all related to the Lempel-Ziv algorithm and to recent algorithms for estimating Hausdorff dimensions. The average storage and running times increase as N and NlogN, respectively, with the sequence length N. These algorithms proceed basically by constructing efficient codes. They seem to be the optimal algorithms for sequences with strong long-range correlations, e.g., natural languages. An application to written English illustrates their use. This is an advanced, but excellently written paper, containing rigorous mathematical analysis. Some effort is needed for implementation and further evaluation for practical use.

36. W. A. Finamore and P. R. L. Nunes, "A Modified Lempel-Ziv Algorithm and its Application to Image Compaction", Proc. of the 1991 International Conference on Acoustics, Speech and Signal Processing, ICASSP'91, vol. 4, Toronto, Ontario, Canada, April 14-17, 1991,pp.2769-2772.

A modified Lempel-Ziv algorithm (m-LZA) for data compaction is proposed. The performance of the m-LZA in terms of compaction rate is better than the performance of the LZA. The improvement which can be understood by examining the tree structure underlying is corroborated by comparing the compaction rates obained with the application of both algorithms to the compaction of image data. Another interesting feature of the m-LZA is its independence of the size of the source alphabet. This paper provides yet another modification of the LZA, one particularly appropriate to image compaction. The question arises whether this modification is appropriate for other data. Evaluation for other data would be interesting.

37. H.U. Khan and H.A. Fatmi, Text Compression Using Rule Based Encoder, Electronics Letters, vol. 30, pp.199-200, Feb. 3, 1994.

A novel compression algorithm has been developed on the principle that optimal compression ratios (CRs) are achieved when new order is perceived at each step of the compressing algorithm. In a previous communication this rule was to the decoding part of the LZ algorithm and a 50% improvement in the CR was observed. In this letter the rule is applied to the encoding part of the compressing algorithm. This resulted in improviment in the CR by more than 230%. This sounds like a promising method which deserves further experimentation and evaluation. While it was developed specifically for text, it might also perform well with some other sources of data.

38. A.D. Clark, "Adaptive Source Coding Techniques for Communications Systems", Proc.Second IEE National Conference on Telecommunications (Conf. Publ. No. 300), York, UK, April 2-5, 1989, pp. 53-60.

This paper discusses the particular constraints that the real time nature of communications systems place on the design of source codes, and presents one particular solution based on a modified LZ algorithm. The design criteria for real time source encoders for communications presented in this paper are:

(1) compression performance;

(2) delay;

(3) compression of non-stationary sources;

(4) avoidance of data expansion;

(5) interaction with the ARQ system; (6) error propagation; and

(7) implementation complexity.

Adaptive variable length source codes, the main families of which are based on Huffman, Shannon-Fano and arithmetic coding, are discussed. Adaptive string encoding algorithms in the two categories of sliding window and dynamic dictionary are described. The first is exemplified by the LZ77 algorithm. The second is exemplified LZ78, LZW and Miller and Wegman's variation of LZ algorithms. In the dynamic dictionary algorithm an indexed dictionary of strings is constructed, and the index of string used as the codeword. This has proved to be a more practical method than the sliding window technique, as it is not necessary to transmit the length of each string (the decoder can deduce this) and methods exist which permit efficient implementation and fast string matching. Commonly, hybrid source coding schemes are used, which embody more than one of the techniques described above. Three such hybrid schemes are described. These are:

(1) concatenation, examples of which are the use of run length encoding prior to Huffman coding, and the concatenation of LZ and arithmetic coding;

(2) initial selection, in which part of the data is initially compressed using several coding schemes in parallel, and the code achieving better compression selected; and

(3) dynamic selection, in which several coding schemes are operated in parallel, and a decision made periodically on which code to select.

The last approach is more appropriate in the case of a communications system than the second one as the source characteristics are likely to be non-stationary. Four commercially developed adaptive source coding schemes were tested off-line, and a comparison made on the basis of performance and and complexity. These were:

(1) a variable length encoding scheme;

(2) a hybrid run length, first order Markov scheme;

(3) a hybrid first order Markov with sliding window; and

(4) LZ, with a simple cyclic scheduling algorithm (CSA) for storage recovery.

Of the design criteria for real time adaptive source encoders memory requirements and processing time are probably the most critical. The LZ algorithm with character extension improvement and a simple storage recovery heuristic provides performance equivalent to first order variable length encoding algorithms, at a fraction of the processing time.

Further work is needed to improve the tolerance of adaptive data compression algorithms to undetected transmission errors, and investigate the synchronization problems that may occur with string encoding algorithms. Comparisons of the compression ratios achieved using the modified LZ method with an ideal variable length code ,and three hybrid schemes are given in tabular form, as well as the comparisons of compression ratios achieved by the modified LZ method for dictionary sizes of 512, 1024, 2048 and 4096 entries, for seven different files in each case This is a very instructive and useful paper, but with no mathematical analysis. It is relatively easy to read.

39. T. Kawabata and H. Yamamoto, "A New Implementation of the Ziv-Lempel Incremental Parsing Algorithm", IEEE Transactions on Information Theory, vol. 37,pp. 1439-1440, Sept. 1991.

Combining an idea published by Rissanen and an idea of enumerative coding, a new implementation of the Ziv-Lempel incremental parsing algorithm for coding and decoding discrete data sequences is obtained. The encoder description is given in program form. The algorithm in this paper is worth exploring because of potential gains due to the incorporation of methods from context modeling and parsing trees.

40. Y. Bar-Ness and C. Peckham, "Word Based Data Compression Schemes", Proc. 1989 IEEE International Symosium on Circuits and Systems, Portland, OR, May 8-11, 1989, pp. 300-303.

Three different source models for word based data compression are proposed: Move to Front, Frequency to Front, and Alpha-Numeric to Front. Their principles and method for encoding the gathered data context are presented. Results of compression ratios obtained are included and compared. Comparisons of performance with those of the Lempel-Ziv algorithm and 4th order arithmetic encoding are also made. Finally, some ideas for further improving the performance already obtained are proposed. This paper presents some techniques which offer fresh alternatives to the many LZ-related approaches and, through further exploration, may offer improvements.

41. S. Yoshida, Y. Okada, Y. Nakano, H. Chiba and M.Mori, "New Binary Image Compression Scheme for Various Kinds of Images", Proc. SPIE Conf. on Image Processing Algorithms and Techniques III, vol. 1657, 1992, pp. 124-128.

This paper presents an efficient lossless data compression scheme for various kinds of binary images such as line drawings and half-tone pictures. A study of a combination of preprocessing and Lempel-Ziv universal coding is described. To improve the compression ratio, the LZ universal coding was modified by dividing its dictionary into classified sub-dictionaries. Improved compression ratios were obtained in computer simulation on composite images consisting of mixed text and images. The techniques in this paper may be useful for wider classes of data. For this reason they should be further explored and evaluated.

42. S. Apiki, "Lossless Data Compression", Byte, March 1991, pp. 309-312, 314 and 386-387.

This paper states that Huffman coding and LZW coding are at the root of most compression. (Not that compaction is the term that should be used because of the lossless nature of the process). Dynamic Huffman encoding and decoding are described, and pseudocode given. The input probabilities do not have to be known for dynamic Huffman coding. The LZW algorithm is described and pseudocode given for both encoding and decoding. Expansions and enhancements are described. This paper contains some useful insights, information and techniques. Furthermore, the pseudocode may be a useful starting point for someone wishing to develop software implementations of these algorithms.

43. N. Ranganathan and S. Henriques, "High-Speed VLSI Designs for Lempel-Ziv-Based Data Compression" IEEE Trans. on Circuits and Systems--II: Analog and Digital Signal Procesing, vol. 40, pp. 96-106, Feb. 1993.

The chip described in this article will operate at a transmission rate of 13.3 million bytes per second. The data compression hardware can be integrated into real-time systems so that data can be compressed and decompressed on the fly. This paper contains state-of the-art information on the high speeds of compression and decompression that are now possible. It is highly recommended for an understanding of what is now available in high-speed hardware.

44. B. Ya. Ryabko, "Fast and Efficient Coding of Information Sources", IEEE Trans. on Information Theory, vol. 40, pp. 96-99, Jan. 1994.

The two cases of known and unknown statistics are investigated. It is noted that all known methods fall under one of these two classes. The Ziv-Lempel codes and their variants fall under the first class, and the arithmetic code and Lynch-Davisson code fall under the second one. The codes from the first class need exponential memory size for redundancy r when r approaches zero. The methods from the second class have a small memory size but a low encoding speed. A code is presented that combines the merits of both classes: the memory size is small and the speed is high. The code can be easily realized by a parallel-sequential method. This is very important for hardware implementation. It also provides good data compression in the case of comparatively small block lengths. This paper contains deep and noteworthy advances in the state of the art.

45. S. Bunton and G. Borriello, "Practical Dictionary Management for Hardware Data Compression" Communications of the ACM, vol.35, pp.95-104, Jan. 1992.

This article addresses the problem of realizing the LZ78 algorithm in hardware. The combined effects of dictionary size and of various dictionary management mechanisms on the performance of a Ziv-Lempel encoder are evaluated and then a new dictionary manager is presented which resolves the time-space trade-offs of its predecessors. Comparisons with commercial chips are given, and it is shown that the new LZ variant compression performance, memory requirements and speed are superior to previous versions. Many graphs of performance are given, as well as algorithms in program form. This is a very good and useful article.

46. T. Bell, I.H. Witten and J.G. Cleary, "Modeling for Text Compression", ACM Computing Surveys, vol.21, pp. 557-591, Dec. 1989.

The best schemes for text compression use large models to help them predict which characters will come next. The actual next characters are coded with respect to the prediction, resulting in compression of information. Models are best formed adaptively, based on the text seen so far. This paper surveys successful strategies for adaptive modeling that are suitable for use in practical text compression systems.

The strategies fall into three main classes:

> finite-context modeling, in which the last few characters are used to condition the probability distribution for the next one;

> finite-context modeling, in which the last few characters are used to condition the probability distribution for the next one; finite-state modeling, in which the distribution is conditioned by the current state (and which subsumes finite-context modeling as an important special case); and

> dictionary modeling, in which strings of characters are replaced by pointersinto an evolving dictionary.

A comparison of different methods on the same sample texts is included, along with an analysis of future research directions. This is still a useful and comprehensive paper, even though it was published back in 1989.

47. A.D. Wyner and J. Ziv, "The Sliding-Window Lempel-Ziv Algorithm is Asymptotically Optimal", Proc. IEEE, vol. 82, pp. 872-877, June 1994.

This version of the LZ77 algorithm is used in the highly successful "Stacker" program for personal computers. It is also incorporated into Microsoft's new MS-DOS-6. Although other versions of the Lempel-Ziv algoirithm are known to be optimal in the sense that they compress a data source to its entropy, optimality in this sense has never been demonstrated for this version.

In this paper the algorithm is described. It is shown that as the "window size", a quantity which is related to the memory and complexity of the procedure, goes to infinity, the compression rate approaches the source entropy. The proof is general, applying to all finite-alphabet stationary ergodic sources.

48. T.A. Welch, "A Technique for High-Performance Data Compression", Computer, pp. 8-19, June, 1984.

This is the often referenced paper in which the extension of LZ to LZW is explained.

49. J.Rissanen and G.G.Langdon, "Universal Modeling and Coding", IEEE Trans. on Information Theory, vol. IT-27, pp. 12-23, Jan. 1981.

This is a fundamental and classic paper. It is frequently referred to.

50. G.G. Langdon, Jr., "A Note on the Ziv-Lempel Model for Compressing Individual Sequences", IEEE Trans. on Information Theory, vol. IT-29, pp.284-287, March 1983.

This is famous short paper which, by describing the context and coding parameter for each symbol, provides insight into how the Ziv-Lempel method achieves compression. This paper is often referred to.in the literature on lossless compression.

3.0 Patent List (U.S. and Canadian Patents)

Thirty-four patents were accessed by computer search. Of these only eight appear relevant to Trisignal's interests. It was noticed that there were references to other patents in the descriptions of these eight which were not discovered during the library searches. Therefore, there are additional patents which may be of interest. Copies of the eight patents mentionned above are being ordered through the CRC library.

1. D. L. Whiting, G. A. George, and G. E. Ivey, "Data Compression Apparatus and Method", U.S. Patent No. 5,016,009, May 14, 1991.

2. G. Langdon, Jr., and J. J. Rissanen, "Adaptive Source Modeling for Data File Compression Within Bounded Memory", U.S. Patent No. 4,494,108, Jan. 15, 1985.

3. A. Lempel and G. Seroussi, "Lempel-Ziv Compression Scheme with Enhanced Adaptation; Dictionary Based Data Compression/Decompression System", U.S. Patent No. 5,243,341, Sept. 7, 1993.

4. S. Henriques and N.Ranganathan, "Method and Apparatus for the Compression and Decompression of Data Using Lempel-Ziv Based Techniques", U.S. Patent No. 5,179,378, Jan. 12, 1993.

5. W. L Eastman, "Short-Record Data Compression and Decompression System", U.S. Patent No. 5,087,913, Feb. 11, 1992.

6. I. A. Shah, "Lempel-Ziv Decoder", U.S. Patent No. 5,058,137, Oct. 15, 1991.

7. H. Chiba, Y. Nakano, Y. Okada, and S. Yoshida, "Data Compression Method and Apparatus", U.S. Patent No. 5,150,119, Sept. 22, 1992.

8.G. A. George, G. Ivey and D. L. Whiting, "Data Compression Apparatus and Method", U.S. Patent No. 5,016,009, May 14, 1991.

4.0 Concluding Comments

Figure 2 shows the relationships among topics covered by the references, excluding the patent literature.

References helpful for answering question 1 include: 9,10,12,13,14,15,18,20,21,22,23,24, 25,26,27,28,29,30,32,33,36,37,38,40,41,42,43,44,45 and 46.

References helpful for answering question 2 include: 14,16,17,19,23,33,37,38,40 and 46.

References helpful for answering question 3 include: 11-15,18,19,21,24-29,31,33-39,41,42,44,47 and 48. Figure 2 illustrates this. The block in the centre of the bottom of Figure 2 also contains some practically helpful references.

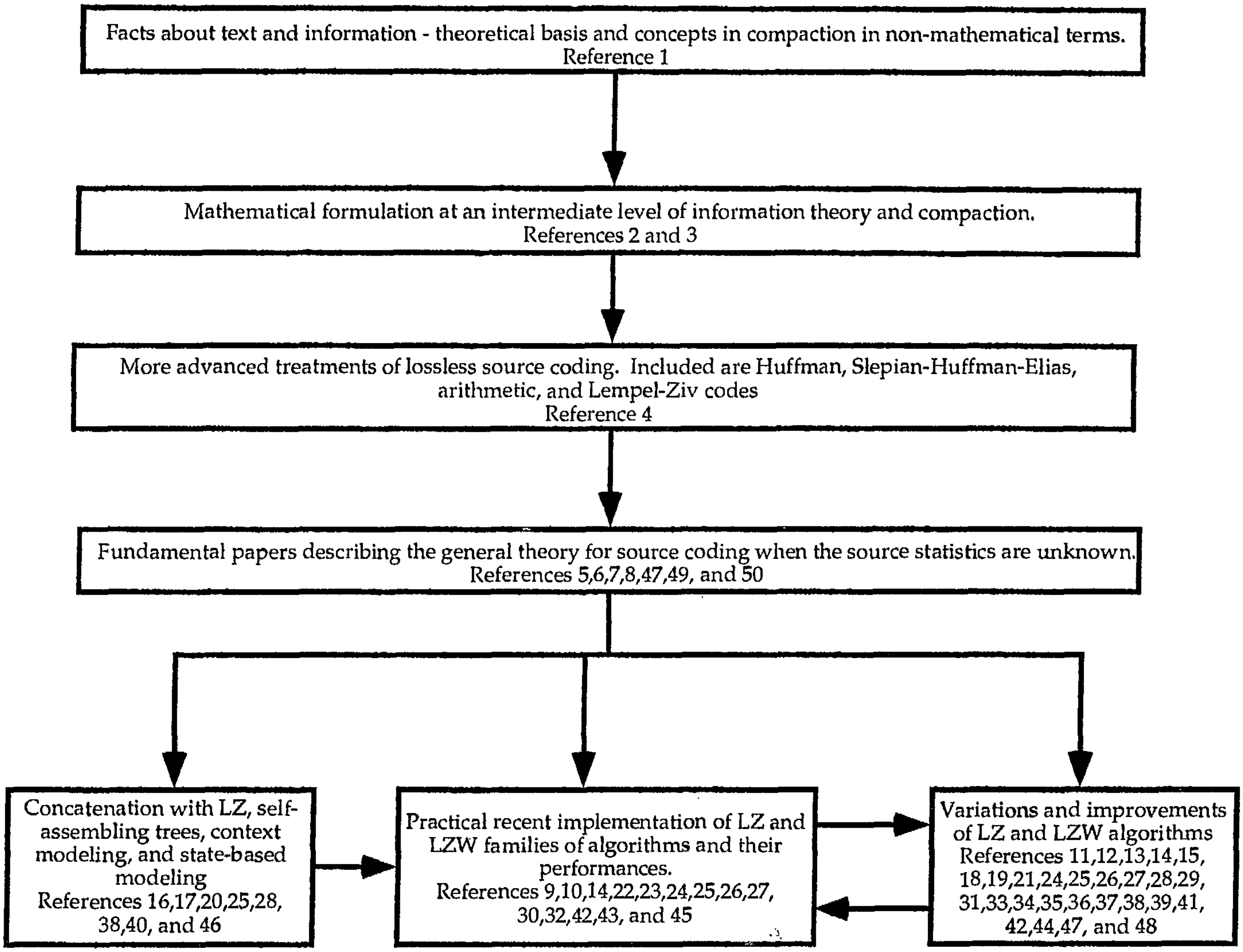

Figure 2: Relationships among the references.

DATE DUE
DATE DE RETOUR
