Norman F. Brickman
Walter S. Rosenbaum

Word Autocorrelation Redundancy Match (WARM) Technology

Word Autocorrelation Redundancy Match (WARM) is an intelligent facsimile technology which compresses the image of textual documents at nominally 145:1 by use of complex symbol matching on both the word and character level. At the word level, the complex symbol match rate is enhanced by the redundancy of the word image. This creates a unique image compression capability that allows a document to be scanned for the 150 most common words, which make up roughly 50% of the text by usage, and upon their match the words are replaced for storage/transmission by a word identification number. The remaining text is scanned to achieve compaction at the character level and compared to both a previously stored library and a dynamically built library of complex symbol (character) shapes. Applying the complex symbol matching approach at both the word and character levels results in greater efficiency than is achievable by state of the art CCITT methods.

1. Introduction

The Word Autocorrelation Redundancy Match (WARM) is an intelligent facsimile method for image compression of textual documents. The technique makes practical the storage and transmission of very high resolution facsimile imagery of text. Briefly, the WARM algorithm works as follows. Both the encoding and decoding phases have available to them a list of the 150 most common words in the English language and the font images of the upper- and lower-case alphanumeric characters. When a sufficient number of rasters has been read in to discern a line of text[1], then the WARM algorithm begins processing by detecting the blanks between words. The 150 most common words in the English language represent approximately 50% of the words in an average composition [2]. The facsimile image of each of the word fields that have been detected on a scanned line is compared to the font image of the 150 most common words. If a word passes the match threshold, then a code for that word replaces the corresponding word facsimile image.

The match on the word level is performed without recourse to character segmentation, and a word can be reliably matched even if its individual characters cannot be segmented and/or matched separately. Hence, the first phase of WARM processes the document at the word level, taking advantage of frequently recurring words and the redundancy (additional information) available at the word level, which is not necessarily available at the character level. For those word fields that fail to be matched with any of the set of the 150 most common words, the WARM algorithm proceeds to search for character segmentation breaks and then attempts to match the characters with the set of alphanumeric pre-stored character images. The pre-stored images are referred to as complex symbols. If WARM fails to discern a match for the segmented video at the character level, the unmatchable video is added to the set of complex symbols at both the encoding and decoding stages of the algorithm. The images of the added complex symbols are compressed using the CCITT two-level MREAD technique [3] or the multi-level technique [4].

The three phases of the WARM algorithm—word, character, and facsimile—are shown schematically in Fig. 1. Nominally, at the word level, a 500:1 compression is achieved; at the character level, a 150:1 compression is achieved; and for those unmatched characters that are sent facsimile (and added to the complex symbol set), a 3:1 compression is achieved. Overall, this results in a 145:1 compression for the average text document and compares

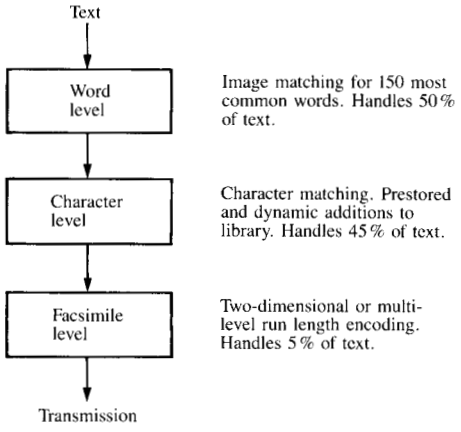

Figure 1 Three levels of operation in WARM.

with a range of 7:1 to 15:1 achieved using the CCITT two-dimensional MREAD algorithm on the same document. The compression achieved for facsimile within WARM is nominally 3:1 because most white space has already been removed when the video of the character or word is isolated. It should be noted that WARM is a "lossy" technique. That is, absolute fidelity with the original input is not necessarily maintained.

2. Systems overview

• *Document scan*

The scanner conforms with the CCITT Group 3 standard of 1728 black and white pixels per line on a 215-mm size A4 document. Vertical resolution is the optional higher density 7.7 lines/mm, or 2156 lines per A4 size document [3]. At present, WARM is run under the IBM CMS Operating System on a System/370 computer. The scanner output is merged into one file on the CMS system, from which the WARM software can access the data one line at a time and simulate the real interface.

• *Line control functions*

Data from the scanner are read one row of pixels at a time into a circular buffer that consists of 65 rows by 1728 bits per row. Scan lines are searched for text by separating each scan line into contiguous segments which are summed and observed across sequential scans. Agreement among adjacent segments on transitions of the segment sums leads to identification of top, bottom, and base line for a printed line of words on a page. The line detection logic also breaks the line into text *versus* picture portions. Only the text portions are processed by WARM. Picture data are handled directly using MREAD or multi-level compression.

Page skew is corrected within the circular buffer based upon line detection data. Continual monitoring occurs as the paper proceeds through the feed-through scanner. With $y(i)$ being a line bottom for each segment $S(i)$, the skew of the paper is obtained by minimizing

$$\sum_i [y(i) - mS(i) - b]^2 \tag{1}$$

for a linear fit to the least squares, where m is the slope and b is the intercept, and the summation is over the n segments i. The slope is given by

$$m = \frac{\begin{vmatrix} \sum_i y(i)S(i) & \sum_i S(i) \\ \sum_i y(i) & n \end{vmatrix}}{\begin{vmatrix} \sum_i S(i)^2 & \sum_i S(i) \\ \sum_i S(i) & n \end{vmatrix}}, \tag{2}$$

where only those segments are used in the calculation that are within preset bounds. Scan lines already stored in the circular buffer are rotated as soon as the slope is determined or changed, and subsequent scan lines are de-skewed upon placement into the buffer.

Segmentation logic is called after a line of text has been detected. The text is segmented into words. Those words not matched with the set of the 150 most common words are then further analyzed to delineate characters. Segmentation between words is simpler and far more reliable than segmentation between characters, which in part leads to the favorable results achieved by the WARM algorithm.

• *Processing order*

If the word matching and character complex symbol matching fail, the system attempts a match with the previously unmatched video segments (believed to be imperfect character imprints) that were dynamically added to the "character image" library. These dynamically added images are called "objects." For all intents and purposes the objects are treated as character images in both the encode and decode phases of WARM.

This processing order, shown in Fig. 1, undergoes a slight reordering if the font of the text is not known *a priori*. Under these circumstances WARM proceeds initially at the character level. Each segmentable character is compared to candidates from the repertoire of font character sets. The font that yields the statistically more significant match rate is concluded to be the document's font. WARM then continues at the word level, generating the respective 150 word images from the assumed font.

• *Preliminary screening*

A screening algorithm is used to limit processing to likely match candidates. Words and characters are screened on the basis of both width and height. Words are further screened using a modification of the surrounding area code technique [5]. The top and bottom portions of scanned words are searched for strong strokes (ascenders and descenders, respectively) and the location of such strokes, as shown in Fig. 2. The distance metric (DM) for comparison of a scanned unknown (u) to a library entry (e) is

$$\begin{aligned} DM(u-e) = {} & w(h)\,|H(u) - H(e)| \\ & + w(w)\,|W(u) - W(e)| \\ & + w(a)\sum_i |A(u,i) - A(e,i)| \\ & + w(d)\sum_i |D(u,i) - D(e,i)|, \end{aligned} \tag{3}$$

where $H(\cdot)$ is height, $W(\cdot)$ is width, and $A(\cdot, i)$ and $D(\cdot, i)$ are the location data for the ith ascender and descender, respectively. The $w(\cdot)$ terms are weights that are used to normalize the distance.

• *Symbol match logic*

The WARM algorithm processes characters and words in the scanned text using a template match [6, 7] and nonlinear difference code summation combined with N-dimensional weighting using prestored feature vectors. A word is matched by overlaying it from left to right with the font image of words from the library of the 150 most common words. No attempt is made to segment characters within the word. This means that, even if a word cannot be broken into separate characters, the word match processing still proceeds.

The process starts with the image of library words being superimposed on a scanned word field. A difference, or exclusive-OR, image $D(i, j)$ is formed,

$$D(i,j) = U(i,j) \oplus E(i,j), \tag{4}$$

where U and E are the unknown and library character pixels at location (i, j), respectively. This is followed by execution of a correlation algorithm which assigns a weight $w_1(i, j)$ to each bit in a contiguous group of bits based upon the number of neighbors. See Table 1. Each contiguous cluster k has a correlation value $S_1(k)$ given by

$$S_1(k) = \sum_{i,j} [w_1(i,j)D(i,j)], \tag{5}$$

where the summation is over contiguous bits.

Figure 3 gives an example of this process.

The correlation values for each exclusive-OR bit cluster are summed for a total correlation, CT_1, which is then used in the determination of the best character shift position,

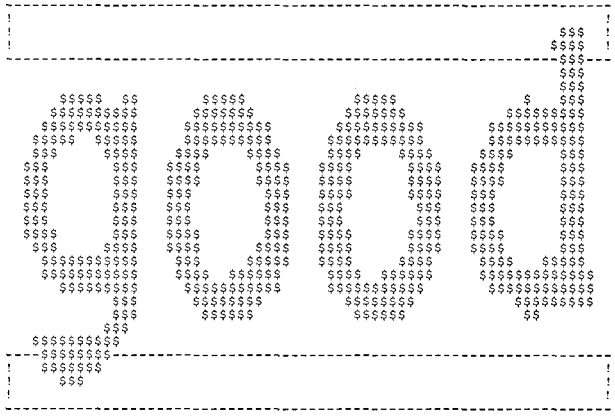

Figure 2 Upper and lower rectangular zones are used in obtaining strong strokes used in word screening.

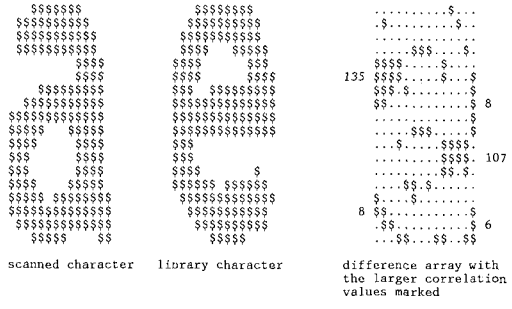

Figure 3 Example of the exclusive-OR difference pattern obtained when a library letter "e" is compared to a scanned "a" and the correlation weights that result.

Table 1 Bit weightings used in quantification of correlation node density.

Number of neighbors	*Weight* $w_1(i,j)$
0	1
1	1
2	2
3	12
4	25

$$CT_1 = \sum_k S_1(k). \tag{6}$$

The library character is first positioned over the scanned image in the expected, or central, position. Other match positions are tested, up to as many as two pixel positions horizontally or vertically removed from the expected position, with the measurement criterion being minimization of the total correlation CT_1.

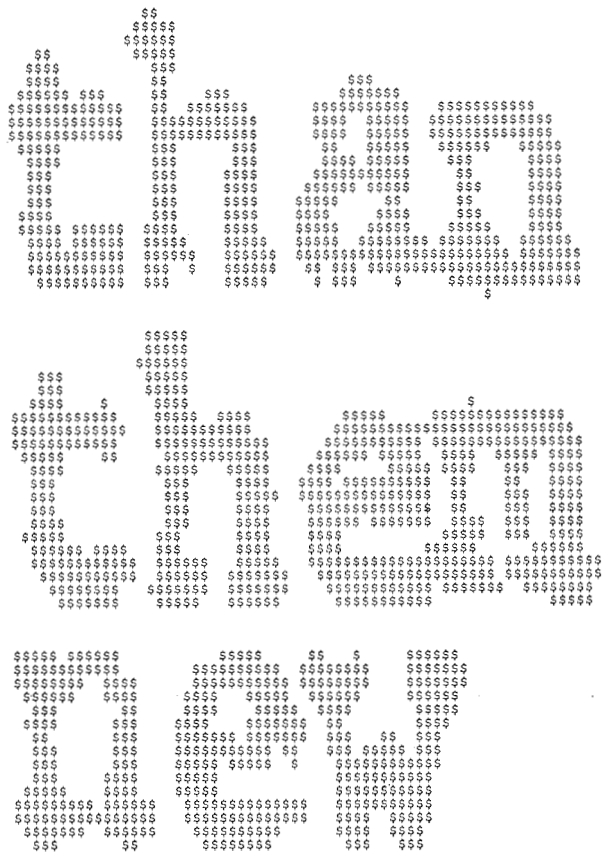

Figure 4 Examples of character-level segmentation problems from scanned Page B. These words were correctly matched at the word level by the WARM process.

Table 2 Results of WARM word match logic applied to scans of Page A. There were no substitutions throughout.

Font	*Page quality*	*Percent word match*
Prestige Elite	Original	95
	2 Reproductions	92
	4 Reproductions	75
Letter Gothic	Original	98
	2 Reproductions	97
	4 Reproductions	94

After the best character position is obtained, a new correlation value $S_2(k)$ is calculated for each contiguous cluster k using weighted feature vectors $w_2(i, j)$ associated with each library character,

$$S_2(k) = \sum_{i,j} [w_2(i,j)w_1(i,j)D(i,j)], \qquad (7)$$

and then a new total correlation CT_2 is calculated:

$$CT_2 = \sum_k S_2(k). \qquad (8)$$

CT_2 improves match discrimination since the natural differences between character shapes are enhanced.

3. WARM simulation and test documents

WARM is presently undergoing computer simulation on a System/370 under CMS. Complete character and word match software has been developed. The program has been implemented in Pascal/VS, with assistance from System/370 Assembler programs in several computation-intensive areas. The Pascal code has been kept relatively universal to assist in migration to other processors in the future. The character library is presently composed of Prestige Elite and Letter Gothic fonts at 12 characters per inch, and Courier at 10 cpi, all generated from an IBM ®Selectric typewriter. Fonts used in the system may be of a fixed pitch or proportional spacing type.

Storage requirements for WARM vary with the number of fonts resident in main memory at one time. Each font requires 3K bytes. The WARM program requires roughly 100K bytes for instruction storage and another 125K bytes for static and dynamic memory during execution.

Two types of textual pages were created and scanned to provide initial testing of the WARM algorithm:

A. Confusable words
B. Library carbon copy words

Page A has the 150 library words plus every legal English language word that differs by one character from the 150 words, for a total of 692 words. Page A was further scanned in its original form as well as after two copier reproductions and after four copier reproductions.

Page B has the 150 library words, typed in Prestige Elite font, in three separate paragraphs on the same page. The first set of words is in normal clear type, the next set is typed through two sheets of carbon paper, and the third set is typed through four sheets of carbons.

Extensive testing of the operation of WARM was conducted by gathering actual documents that have been generated or received by several of the departments at our location.

4. System test results

Page A was tested in Prestige Elite 12 font and Letter Gothic 12 font. Word match was never lower than 75% and went to 98% on original copy. There were no word substitutions. See Table 2.

The original copies of Page A were also tested for character matching. Of those words not matched by WARM, statistics were kept on the number of segmented video symbols matched with prestored library characters *versus* those that are candidates for matching with the dynamically added library objects. Table 3 shows the predominance of matches with the prestored character library.

Page B results demonstrate the WARM approach under very adverse conditions. The page was copier-reproduced once followed by a scan that was inadvertently too dark, causing characters to lose sharpness and bleed into one another. The words shown in Fig. 4 are examples of images of scanned words from the two carbon area of Page B that were correctly matched as words even though the system could not have segmented the words into characters. The words in the two-carbon area of Page B, with quality as shown in Fig. 4, were matched 64 percent of the time. Words in the four-carbon area of the page were matched 9 percent of the time. There were no word substitutions.

Representative samples of results obtained from testing a set of 30 documents obtained from actual office correspondence are summarized in Table 4. The percent word match numbers are based upon those words in the documents that are in the WARM library and hence are eligible to be matched. Again, there were no word substitutions.

Font detection statistics have consistently shown very peaked response characteristics, demonstrating a rapid and accurate discrimination of the font being scanned. With multiple fonts loaded in the WARM memory, typical results give a 90 percent character match rate in only the single correct font and less than 0.1 percent correct character match in an incorrect font. The remaining character matches are for the same character matched in multiple fonts.

5. WARM system implementation

A standard size 8½ × 11-inch average page, which hypothetically has 400 words with an average length per word of five characters, can be represented in 2.4K bytes. This amounts to a 194:1 compression of the document relative to facsimile, which requires 3.7M bits or 466K bytes. The CCITT MREAD compression ranges from 7:1 to 15:1, depending upon the density and sharpness of the text, and implies between 30K and 66K bytes per page.

An intelligent facsimile device can use the WARM technology in two ways. The first uses words, characters, and facsimile, as shown schematically in Fig. 1. This gives a compression rate of 145:1 or 3.2K bytes for the above document based upon average transmission rates of 16 bits per word, 12 bits per matched character, and 100 bits per compressed facsimile character. A second mode of WARM implementation involves the use of word and facsimile levels, eliminating the intermediate character mode. Words that are not matched (50% of the total) are sent using multi-level or CCITT MREAD, giving a compression rate of 36:1 or 12.8K bytes (Table 5).

Table 3 Results of WARM character-level processing of unmatched words on Page A. The unmatched words were segmented into primitive symbols and matched with candidates in the prestored character library.

Font	*Page quality*	*Percentage of symbols matched*
Prestige Elite	Original	88
Letter Gothic	Original	97

Table 4 Representative results obtained from testing a set of actual office correspondence documents for word match operation of the WARM system.

Font	*Page quality*	*Percent word match*
Prestige Elite	Good	89
	Fair	63
Letter Gothic	Good	90
	Poor	54
Courier	Very good	98
	Fair	95

Table 5 Projected WARM system performance (compression) for the two methods of implementation relative to other systems.

Transmission technique	*Compression rate*	*K Bits*	*K Bytes*
Raw facsimile	1:1	3725	466.0
EBCDIC	194:1	19	2.4
CCITT MREAD	10:1	373	46.6
WARM word level	36:1	103	12.8
WARM word/character level	145:1	26	3.2

6. Conclusions

The efficacy of word level match *versus* character level operation has been demonstrated. Overall compression efficiency of 145:1 can be projected from results to date.

7. Acknowledgment

The authors acknowledge the following valuable assistance. Jerome S. Shipman, IBM Federal Systems Division, provided advice during the preparation of this paper. Mark R. Laff, IBM Research Division, assisted with questions that arose on the Pascal language during the programming of the WARM technology. J. Phil Baca, IBM CPD Office System Laboratory, and David F. Bantz, IBM Research Division, have provided all the document scans that have been used in the WARM tests to date.

References

1. Davey L. Malaby, "Scan Centering Device," U.S. Patent 3,506,807, April 14, 1970.
2. H. Kucera and W. N. Francis, *Computational Analysis of Present-Day American English,* Brown University Press, Providence, RI, 1967.
3. Roy Hunter and A. Harry Robinson, "International Digital Facsimile Coding Standards," *Proc. IEEE* **68,** 854–867 (July 1980).
4. M. A. Reed, W. S. Rosenbaum, and A. R. Tannenbaum, "Multilevel Image Compression Algorithm," *IBM Tech. Disclosure Bull.* **24,** 1605–1606 (August 1981).
5. K. Sakai, S. Hiarai, T. Kawada, S. Amano, and K. Mori, "An Optical Chinese Character Reader," *Proceedings, Third International Conference on Pattern Recognition,* 1976, pp. 122–126.
6. R. N. Ascher and George Nagy, "A Means for Achieving a High Degree of Compaction on Scan-Digitized Printed Text," *IEEE Trans. Computers* **C-23,** 1174–1179 (November 1974).
7. W. K. Pratt, P. J. Capitant, W. H. Chen, E. R. Hamilton, and R. H. Wallis, "Combined Symbol Matching Facsimile Data Compression System," *Proc. IEEE* **68,** 786–796 (July 1980).

Received April 5, 1982; revised June 11, 1982

Norman F. Brickman *IBM Communication Products Division, 18100 Frederick Pike, Gaithersburg, Maryland 20879.* Dr. Brickman is an advisory engineer in the Advanced Office Systems Technology Department. He is presently involved in software development for office facsimile systems. He joined IBM in 1969 at the Poughkeepsie, New York, development laboratory in the FET Memory Development Department, where he worked on the design and development of memory and logic semiconductor devices. In 1973 he began work on system design procedures and design automation associated with the use of programmable logic arrays, and sensor-based computer applications. From 1976 to 1981 he had responsibility for the architecture, design, and associated LSI development for a part of the IBM 7880 satellite communications controller developed under subcontract for Satellite Business Systems. This work involved development of programmable logic array and microprocessor based logic subsystems. He received a B.S. in electrical engineering in 1963 from Johns Hopkins University, Baltimore, Maryland, and an M.S. and a Ph.D. in low-temperature physics from Yale University, New Haven, Connecticut, in 1964 and 1969. Dr. Brickman is a member of the Association for Computing Machinery and the Institute of Electrical and Electronics Engineers. He received an IBM Outstanding Innovation Award in 1977 for his work in programmable logic arrays and has received the second plateau invention award.

Walter S. Rosenbaum *IBM Communication Products Division, 18100 Frederick Pike, Gaithersburg, Maryland 20879.* Dr. Rosenbaum is the manager of the Advanced Office Systems Technology Department. He was a major participant in the development of new information handling technology for IBM office systems and products. His group has implemented the basic spelling feature technologies that are used in Displaywriter, DOSF/8100, SCRIPT/VS, PROFS, and Datamaster. Dr. Rosenbaum joined IBM from the National Aeronautics and Space Administration in 1969. From 1969 to 1972 he worked in the Federal Systems Division's earth resources/ecology new business area developing mathematical hydrologic models and statistical technology for forecasting river flow from space photography. From 1972 to 1975 he was involved in the development of multifont postprocessing error correction for the postal service advanced optical character recognizer (AOCR). Dr. Rosenbaum received a B.S. in mathematics from Yeshiva University in 1966, an M.S. in applied mathematics from New York University in 1968, and a Ph.D. in mechanical engineering (theoretical thermodynamics) from the Catholic University in 1972. Dr. Rosenbaum has received two IBM Outstanding Invention Awards and the seventh patent plateau award.