R. L. Griffith

Data Recovery in a Photo-Digital Storage System

Abstract: A data-recovery feature has been developed for recovering electron-beam recorded information which is microscopic in dimension and has been partially obliterated by flaws in a photographic-film recording medium. This feature provides (1) backup procedures that exploit redundancies in the recording format for the synchronization and identification of data, (2) coding for error detection and correction of 5 independent characters in 50 data character lines, (3) variation of machine parameters that affect reading performance, and (4) statistically optimized schedules for applying a variety of recovery techniques. Error rate is reduced from one error line in about 100 lines to less than one error line in 2.7×10^6 lines.

Introduction

The theory of error detection and correction codes is finding its way into practice. With these codes, machines can reconstruct data that have been destroyed. The first practical implementation of a powerful and complex code was achieved in the trillion-bit IBM Photo-Digital Storage System,[1] which uses a Reed-Solomon code for correcting up to 5 error characters in a line of 50 data characters.

With a correction facility able to correct bursts of errors in a line, a recorded bit need not be larger than the average flaws in the recording medium. In this system flaws cannot be eliminated by pretesting the medium, because the recording is permanent on silver halide photographic film. However, since the film has high resolution, there is a trade-off between the cost of the correction facility and the cost savings of high-density recording.

The data-correction facility alone is not adequate. A line can be corrected only if each character read is properly identified by its position on the line (first, second, third, etc.). The line to be read must also be identified. Character and line locators are recorded with each line, but the locators are subject to the same disturbances as the data.

This paper describes methods for quickly overcoming disturbances to the character and line locators. Should these methods fail, there are slower backup procedures that use the error-correction facility. Varieties of these and other recovery tools are applied to each problem line. The schedule of tool application is based on a statistically optimized strategy.

The author is located at the IBM Systems Development Division Laboratory, San Jose, California 95114.

System performance with the data-recovery procedure is also reported. Statistics relative to recording quality and data recovery are gathered by a dedicated control processor which runs the mass-storage system.

In a system where actual reading is in process 30.6% of the time, only 0.67% of the system's time is spent on all data-recovery procedures.[1]

Storage medium and data format

The storage medium is a 1.38 by 2.75 inch photographic film called the "chip," illustrated in Fig. 1. It is coated with an emulsion specially designed for electron-beam recording. The chip is recorded and developed on-line.

Nearly five million bits of data can be recorded on one chip. Thirty-two chips are stored in a plastic box called a "cell" (Fig. 1). In the trillion-bit configuration, about seven thousand cells are stored on-line, any one of which can be accessed in less than three seconds. During storage and pneumatic transport, the cells are closed to keep the chips free from dust.

The format of the information recorded on the chip involves a combination of base lines, data bits, line format, frame format, and line and character locators. Each is briefly described in this section.

- *Base lines*

Figure 2 shows a sample of recorded data. The data lines are separated by alternate transparent and opaque bands called "base lines." In the photo-digital reader, these base lines guide a spot of light along a data line. If the spot drifts toward a transparent base line, the average level of light passing through the chip increases. This is a signal for a flying spot scanner servo to steer the

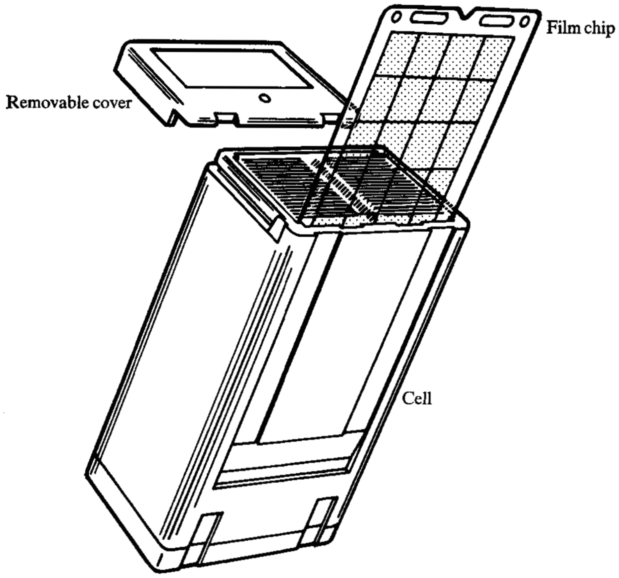


Figure 1 Chip and cell.

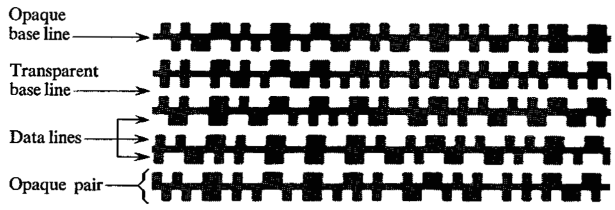
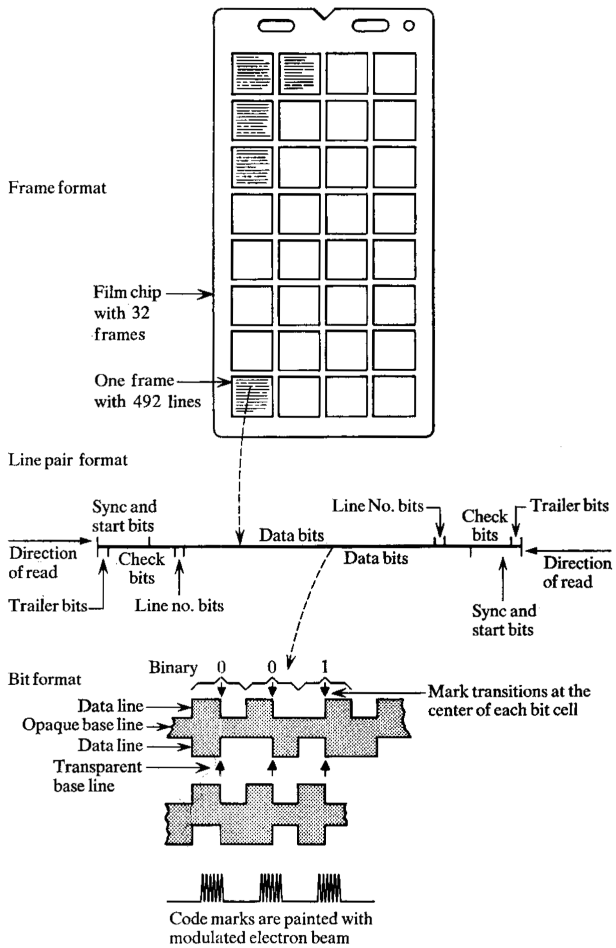


Figure 2 Recorded data.

Figure 3 Chip and data format.



spot back toward the data line. Similarly, a drift toward an opaque base line produces a signal of opposite polarity, which also causes the flying spot to return to the data line. At the end of a line, controls deflect the spot to an adjacent line to scan back in the opposite direction. The separation between data lines also prevents cross-talk effects in the reading process.

• *Data bit representation*

The data line is a series of marks, each 8 microns square. A mark is either wholly transparent or opaque. Figure 3 illustrates the pattern of opaque marks "painted" onto film by the electron beam. Transparent marks are "unexposed" areas on the film.

As seen by the scanning spot, a "1" bit is encoded as a transparent mark followed by an opaque mark, and a "0" bit coded vice versa. This system of encoding, known as phase modulation, was selected over frequency modulation (like marks for "0", opposite marks for "1") because its bit cells are independent and it provides more definite bit detection. The polarity of the signal for a bit does not depend on the nature of the previous bit, and the difference between bit and flaw shapes is exploited. Integrating the total light intensity over the bit cell collects all the available recorded information for that bit, as shown in Fig. 4. An all-opaque or all-transparent cell, due to a chip flaw, is easily distinguishable from a valid bit, and randomly produces either a "1" or a "0." A mottled bit cell often produces the same random result.

Each bit cell has a mark transition (an edge between marks) at its center. It is called a "clock transition" and helps to identify the location of the bit cell.

• *Line format*

The data line is segmented into 6-bit characters, indicated in Fig. 3, for purposes of error correction. First in the data line is a fixed line-start pattern which is a series of 25 "1" bits followed by the pattern 00001000100101*d*, where *d* is the first bit in the first data character. Following are 50 consecutive data characters, a 2-character line number, and 11 redundancy characters.

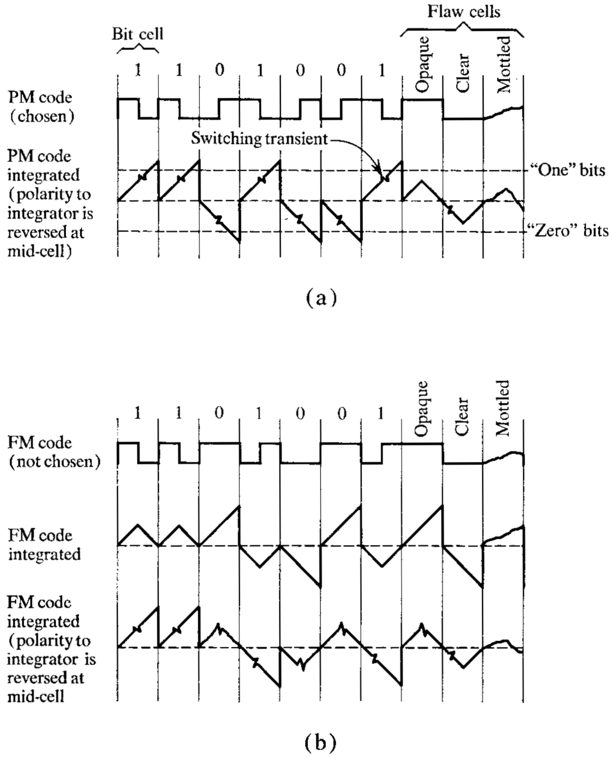


Figure 4 Encoding scheme. (a) The flaw cells do not register as valid bits. (b) In this code, opaque and clear cells are not distinguishable from valid 1-bit or 0-bit cells.

• *Frame format*

The film chip is partitioned into 32 frames. In each frame there are 492 lines (Fig. 3). Even-numbered lines are read from left to right and odd-numbered lines from right to left. The 1's bit in the line number is not recorded, but it is understood from the direction of the reading sweep. Consequently, two lines adjoining one opaque base line have line numbers which appear identical. These are referred to as an "opaque pair."

• *Character and line locators*

Data in a frame could be located by mechanical precision or by reference to locator information recorded with the data. We chose the latter because it eliminates many engineering problems.

In the two-dimensional frame, the scanning spot position in one dimension is indicated by a line locator and in the other dimension by a character locator. The line locator is the base line structure with the data line numbers. The character locator is the start pattern with countable bit cells, counting 6 bits per character. Bit cells are counted by the clock transitions at the center of each cell.

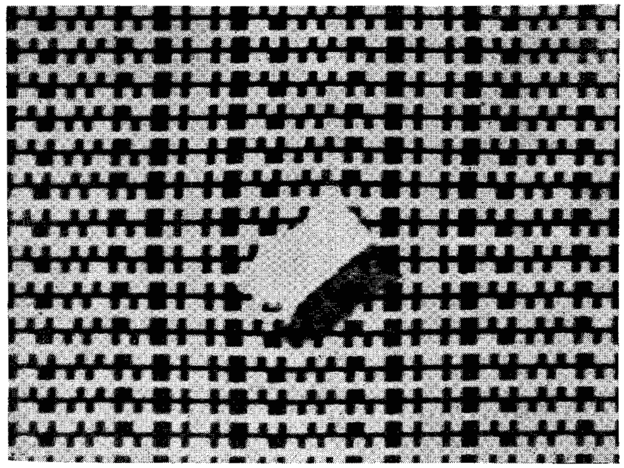


Figure 5 Result from a contaminant which moved (see text).

Chip flaws

Four main types of chip flaws occur. They are caused by contaminants, mechanical marring, emulsion shift, and electrical noise during recording.

A contaminant present during recording prevents the electron beam from properly writing on the sensitive photographic-film chip. If the contaminant is subsequently moved (during chip development, for example), a clear spot remains on the film, shown in Fig. 5. Contaminants which remain after development or are picked up in the developing process or afterwards, usually appear opaque, as indicated in Fig. 6. Liquid droplets which leave a residue produce a mottled effect (Fig. 7).

Figure 8 shows the appearance of surface scratches produced by most kinds of mechanical marring. These are caused by contaminants lying between layers of raw film in roll form and by feed rolls during the coating and cutting of roll stock into chips. Scratches parallel to the data lines are more difficult to correct than others. Mechanical marring of the developed chip has not been a problem, since there is normally no mechanical contact with the recorded surface area in the processes of storage, handling, or reading.

Emulsion shift results from uneven drying in the chip development process. The emulsion gel shrinks as it dries. If a relatively sharp boundary between drier and wetter gel is formed, the gel is pulled toward the dry region. This phenomenon is difficult to detect by visual inspection. It is, however, a minor problem because wetting agents are used in the rinse fluids to reduce such boundaries.

Electrical noise causes nonuniform recording of lines and bits. An example is shown in the center of Fig. 9.

Error detection and correction

A Reed-Solomon code is used to correct up to 5 error characters per line. If there are 6 or more error characters,

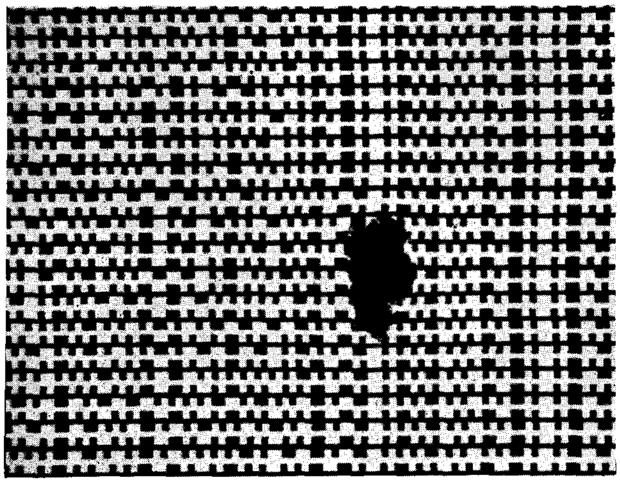


Figure 6 Result from a contaminant which remained in place.

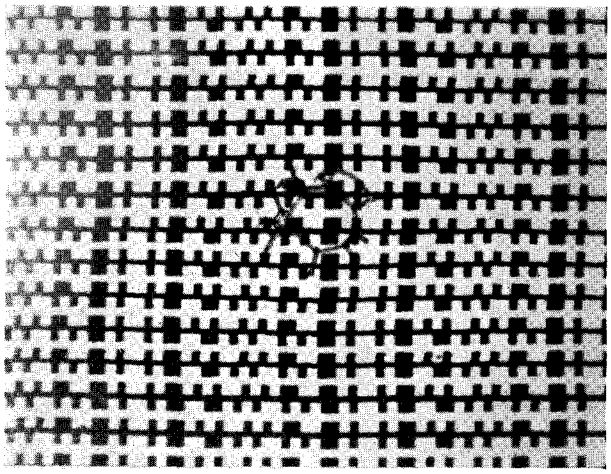


Figure 7 Translucent mottled effect.

no correction can be done on that line. Encoding and error detection are implemented in hardware, and error correction is accomplished by control processor software.

• *Encoding*
The entire line (except for the start pattern) is considered to be a code word. All 63 characters of the line are treated as coefficients of a polynomial in X. The first character is the coefficient of X^{62}, the next of X^{61}, and so on to the last, a coefficient of X^0. Prior to coding, the last 11 characters are all zero. This data polynomial is divided by a generator polynomial, such that there is an 11-character remainder. This remainder is subtracted from the data polynomial to obtain the encoded polynomial. Subtraction is done by substituting the 11 remainder characters for the 11 zero characters in the data polynomial.[2] With the remainder subtracted, the encoded polynomial is an exact multiple of the generator polynomial, and qualifies as a code word. The 63 coefficients of the encoded polynomial are recorded as a data line.

• *Detection*
The new line that results from a disturbance to the encoded line is not likely to be a multiple of the generator polynomial. To test if the encoded line is still a multiple of the generator polynomial, the division process could again be used. However, another method is just as easy, and produces results useful to the error-correction facility.

Note that, if the encoded polynomial is a multiple of the generator polynomial, the roots of the encoded polynomial include those of the generator polynomial. Each of the 11 generator polynomial roots is substituted for X in the possibly disturbed encoded polynomial. Each

Figure 8 A scratch.

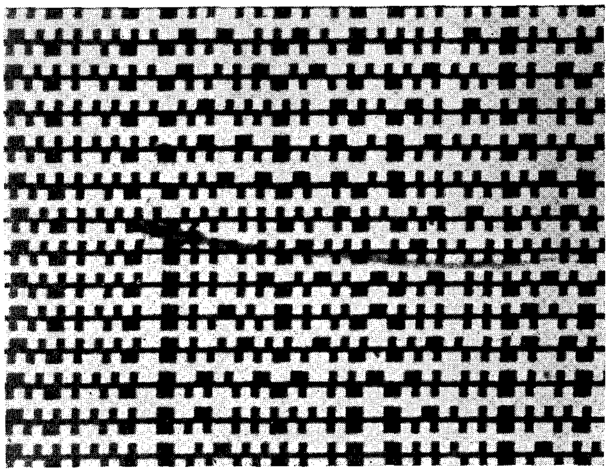
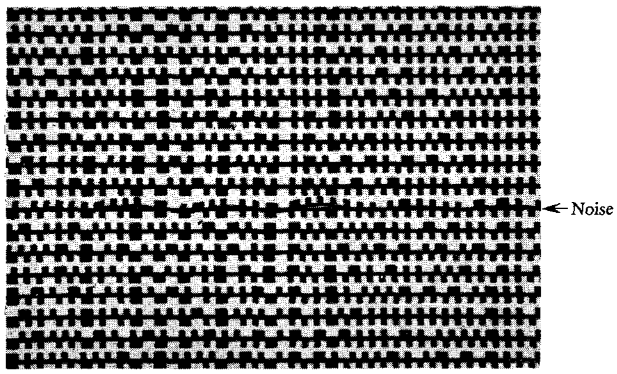


Figure 9 Electrical noise.



and every substitution will produce a polynomial equal to zero if and only if the encoded polynomial is still an exact multiple of the generator. If any result is not zero, an error is detected.

• *Correction*

There are 11 single-character results of root substitution containing information as to *where* and *what* the errors are. These 11 characters are transmitted to the control processor memory, where a program solves up to 10 equations in 10 unknowns. The 10 unknowns are 5 "wheres" and 5 "whats" to allow correction of 5 errors. The 11th resulting character provides parity-like information to strengthen the error-detection capabilities.

Usually, there is only one error and its correction is referred to as 1-EDC (Error Detection and Correction). In this case, it is necessary only to solve two equations in two unknowns (one "where" and one "what"). A special single-error routine is tried immediately following error detection. The results are available, the correction made, and the repaired line retested for error in less time than it takes the scanning spot to move to the point at which it starts to read the line again (168 μs). If the correction is successful, the line is passed over and the following line is read. If not, the line in error is reread.

Similarly, it is expedient to try to correct only 2 errors, then 3, 4, and lastly 5. These attempts are called 2-EDC, 3-EDC, 4-EDC, and 5-EDC, respectively.

The code has been designed to be powerful in error detection. This power is used to guarantee the validity of every error-correction attempt. It also permits the use of many re-READ attempts to help correct the line. Even though each attempt involves a risk of making an erroneous correction, the detection facility more than compensates. The particular generator polynomial used, its selection, implementation, and performance have been reported in an earlier publication.[1]

Line locator problems and recovery techniques

After a cell is transported to the photo-digital reader, the desired chip is extracted from the cell and positioned in a reading window so that an area of about 125 lines is exposed. The scanning spot begins its sweep motion near the center of the window and finds a line. At the line's end, the scanning spot turns around and starts scanning an adjacent line, as indicated in Fig. 10.

If a line-start pattern is detected, the line is read. Its line number, as it appears without correction, is compared against the desired line number on a high-low-equal basis. At the end of the line, a normal scanning spot turn-around occurs when the servo reverses the sweep direction and deflects the spot to an adjacent line. The direction of each line-to-line deflection is determined by the result of the respective compare operation. If all line

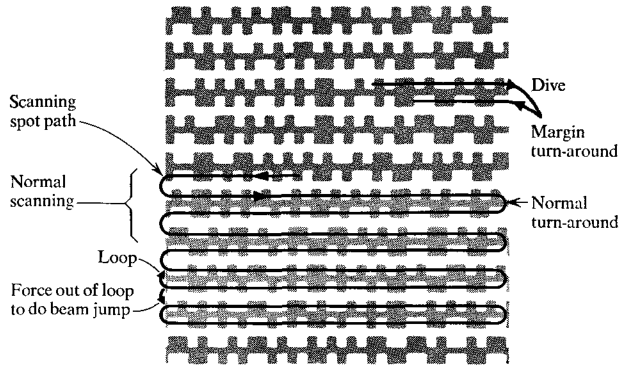


Figure 10 Scanning pattern.

numbers are read correctly, the spot progresses toward the desired line.

When a line number compare-equal occurs, that line is checked for errors by using the recorded redundancy characters. If the line is errorless,[3] it is sent to the host processor, the digital processing system which is using the photo-digital storage system. If the line has errors that can be corrected, the line number in its corrected version is compared again against the desired line number. If it is still equal, the data are sent on to the host processor and reading continues to the next line.

• *Line locator problems*

The problems associated with line locators are hard or soft line-number distortion, hard or soft line-fall-through, a disturbed margin turn-around, and a line-out-of-window.

Line-number distortion. A hard line-number distortion consistently produces the wrong compare result.[4] This can turn the beam back to the adjacent preceding line so that it loops endlessly on that line pair. Soft line-number distortion produces a random compare result; that is, the spot may loop for awhile, but eventually a correct compare result occurs and the line-search process continues.

Line fall-through. Line fall-through occurs when a base line is distorted, especially by a mottled or translucent chip flaw which is not easily detected by the reader's circuits for fall-through prevention. Without the directional guidance of the base lines, the spot may drift to another line and finish the sweep on it. Looping occurs when the scanning spot falls back to the line from which it just came. Only a hard fall-through can cause persistent looping.

Margin turn-around. A margin turn-around occurs at the end of a line on which a line-start pattern is not detected. With no line-start, the location of the end of the line is not precisely known, and the normal turn-around signal cannot be given to the scan control servo. The reader then depends on detection of the transparent margin to signal a turn-around. As soon as the scanning spot enters the margin, it "dives" (Fig. 10) as if trying to correct for a drift into a transparent base line. Margin detection reverses the direction of the sweep and the polarity of the line-servo error signal. As a result, the dive is reversed and biased enough to come out of the margin onto the next line. If the margin has a chip flaw, margin detection can be advanced or delayed; the dive trajectories may also be disturbed. These effects can cause the turn to come out on the wrong line. Again, if this phenomenon is consistent, it is called "hard" because it can cause continued looping.

Line-out-of-window. A line-out-of-window is a functional problem. It implies that either the frame of the chip is out of registration, or that the chip-positioning servo has made an error. In any case, the scanning spot searches toward the edge of the window, looking for the desired line. If it approaches the edge of the window in the forward reading direction, the chip automatically advances; if it approaches the back edge of the window, it is stopped and forced to loop on a pair of lines. The reader automatically attempts to restart the spot near the center of the window, but after four or five such attempts, a 20-ms timeout generates an interrupt to call the attention of the control processor to the problem. This condition is indicated to the control processor so that it can take corrective measures.

• *Line locator recovery techniques*

Three techniques are used for line locator recovery: coarse compare, line jump, and chip repositioning.

Coarse compare. The coarse compare feature is implemented in hardware. It makes use of the fact that the two line numbers in an opaque pair are equal and physically separated, as indicated in Fig. 3. A single chip flaw is not likely to disturb both numbers.

A counter is incremented each time the search process encounters a reversal of scan turn direction. A count of six is taken to mean that the scanning spot is looping, i.e., the failure is deemed by the program to be a "hard" error. (Reversals may not be consecutive because of the possible randomness of marginally read line numbers.) On the assumption that "hard error" looping is occurring, the hardware operates as follows:

(1) Forces the spot to loop on one opaque pair of lines and to include in this loop the line of the original loop which is in the same sweep direction as the desired line.
(2) Allows either line number in the opaque pair to compare with the desired line number.
(3) Considers the line found and worthy of correction if a compare-equal occurs.
(4) Generates an interrupt signal to the control processor if a compare-equal does not occur. A sense point called "reader cycling" indicates the condition.

When coase compare is not effective, it is assumed that too many line numbers in the vicinity of the desired line are wrong or random. The line-jump feature is then needed to find the correct line.

Line jump. The line jump is a hybrid hardware-software feature. It makes use of the line-following ability of the scanning spot and the fact that line numbers of adjacent line pairs are numerically consecutive.

The jump is accomplished by the following sequence. First, the spot is forced to loop on a line pair. A line in that pair is read, and corrected if necessary, to get its line number. The numeric difference $\pm D$ between this line number and the desired line's number is computed by the control processor. This difference provides relative distance and direction information for the jump. The control processor can then issue a command which causes the spot to jump from the line pair it is on to an adjacent line pair. After D such jumps toward the desired line, the control processor assumes it has arrived, and commands a READ.

Chip repositioning. Procedures using line jump can be ineffective in situations involving hard fall-through or disturbed margin turn-around. Each jump can be undone by a fall-through in the opposite direction, for example. Such may be the case if jumps have been tried unsuccessfully a number of times. To get to the other side of the flaw, the chip is moved forward in the window and the search starts afresh on the other side of the flaw. It is assumed that there is only one flaw, and that it covers less than ten lines. The problem may also be encountered while searching in the backward direction, but the occurrence is rare and, therefore, no direct remedial provisions have been made. Several repositionings eventually produce one of the back-up type to be described next; this one back-up type will resolve the rare occurrence.

The backward chip-repositioning remedy is provided for the case where the line is out of the window in the backward direction. When this occurs, the reader generates an interrupt signal. The control processor senses the condition, and intervenes to preempt normal chip servo control and to back up the chip. (The information in the digital servo's positional register is corrected accordingly by the control processor.)

Character locator problems and recovery techniques

A flaw in a series of data bits can obliterate a series of clock transitions. This is compensated for by a tracking oscillator that synchronizes itself with the frequency of clock transitions. When up to 10 clock transitions are destroyed, the "inertia" of the oscillator will carry on bit counting at the same frequency. On the other side of the flaw, clock transitions reappear, and the clock oscillator resynchronizes with them if some drift has occurred.

There is redundancy in the line-start pattern so that up to three bits in error can be tolerated. (The pattern decode hardware is a majority gate.) Should the line-start pattern fail to be detected, it is assumed that more than three bits in the pattern are in error. Consequently, another mechanism is needed to start the reading of the data field. This mechanism is called the "auxiliary line-start" (ALS).

- *Auxiliary line-start*

In 1-, 2-, and 3-error correction, there is a satisfactory degree of certainty that the correction is a valid one, even if there was no confidence in line synchronization. That is, if the line read is started on the wrong bit so that all data is shifted n positions, the probability of achieving an indicated success in correction is very small. If the error-correction facilities indicate that the line is corrected, it is almost certain that it has been read starting with the first data bit. For 4-error correction, this condition is marginal; for 5-error correction, it is not at all satisfactory.

This error-detection ability permits a trial-and-error approach in finding the first bit of a line where the recorded line-start pattern is not detectable. From timing and scan-control considerations, the location of the first data bit is known ± 5 bit positions. An attempt is made to read the line starting from each of the 11 most likely starting positions. When the error detection and correction facilities indicate that the line just read is correct or corrected, this means that the correct position has been used.

A sense bit is provided to the control program to indicate when a line just read was started with ALS. Sensing this condition, the processor uses exclusively the 3-error correction facility. There might be fewer than 3 errors, but as only 1 of 11 ALS READ's can be correct or correctable, it is wasteful to use 1-, 2-, and then 3-error correction as is normally done.

- *ALS for nonline-start*

Another tool for overcoming a distorted line-start pattern is rereading. Whenever the control processor commands that a line be read or reread, the hardware first tries rereading the start pattern. Should the start pattern fail to be detected (nonline-start), the hardware automatically attempts to read the line again, this time using ALS.

A hardware perturbation counter determines which of the 11 possible start locations will be used in the ALS. The first ALS on a given line uses the position that was successful the last time ALS was used. Each successive use on this line will increment the perturbation counter so that, in 11 ALS attempts to read the line, all of the most probable start positions are tried.

- *ALS for false line-start*

A false line-start occurs when flaws in the vicinity of the line-start pattern produce bits which satisfy the majority-gate pattern detector, so that reading starts on the wrong bit. There is no direct detection of a false line-start, other than that the line seems to be unreadable (uncorrectible).

After a number of attempts to read the line have failed, there may well be a false line-start and it is considered worthwhile to force the operation of the ALS feature by means of a command from the control processor for each READ attempt of that type.

Other problems and recovery techniques

Recovery techniques change the conditions under which reading takes place. Three such procedures are parameter variations: offset scanning, extended coasting, and harden clock synchronization. These variations can compensate for differences in reader adjustments, as well as special conditions on the recorded media. The fourth technique is chip sweep, which attempts to remove contaminants (dust) from the chip surface.

- *Offset scanning*

Normally, the scanning spot travels along the data lines so that its geometric center moves parallel to and halfway between a transparent base line and an opaque base line. Any divergence from this path is generally accompanied by a notable increase in single-error correction activity. However, some marginal lines can be read only if the spot is offset "high" or "low" from the center position. Control points allow the control processor to force the spot to ride high or low with respect to the opaque base line.

One reason for using offset scanning is that the flaw looks slightly different at the offset position. Another reason is that very dark or transparent flaws tend to steer the spot off course somewhat before the effect is electronically detected and the beam is switched to the coasting mode. This can be compensated for by the high or low offset in the opposite direction.

- *Extended coasting*

Circuits in the reader can often detect the presence of a "hole" in the medium. Such detection suspends the base-line-following process for a fixed length of time. The spot continues for a fixed distance on a straight-line trajectory in the direction of spot travel prior to suspension. This movement is called "coasting." This often causes the hole to look larger, but it prevents fall-through. Coasting time

Table 1 Data recovery problem and tool list.

Problems motivating data-recovery activity	*Data-recovery tools with respect to problems*
Data errors	Error detection and correction
Line-number distortion	Coarse compare
	Forced looping
	Line jump
	Line-number compare after correction
Looping conditions	Chip repositioning
	Line correct, force forward
Nonline-start	Automatic auxiliary line start
False line-start	Forced auxiliary line start
Fall-through	Extended coasting
Emulsion shift	Hard sync
Improper turn-arounds	Bidirectional line approach (via jumps: arbitrary, direct, and forced)
Dust	Chip sweep
Chip servo misposition and frame misregistration	Chip repositioning

is adjusted for maximum frequency of success. As the coasting time is made longer, there is less error correction capacity for the remainder of the line.

There can be some large holes in a line that is otherwise nearly perfect. Extended coasting can be commanded by the control processor so that the larger hole is crossed successfully. Error correction is then attempted and is often successful.

• *Harden clock synchronization*

Emulsion shifts cause an apparent sudden shift in the data rate. If this is severe, the clock transition tracking oscillator may not follow the shift, and synchronism is lost. The servo gain for the tracking oscillator is adjusted for maximum average success.

Occasionally a difficult line can be read by increasing the tracking servo gain. This is called "harden clock sync," and can be commanded by the control processor to be a condition for the next READ attempt.

• *Chip sweep*

The chip is held between two air-bearing heads which are frames of the reading window in the photo-digital reader. The air jets from these heads are shaped to provide an air-brush effect as well as to support the chip. Contaminants picked up by the chip, after the development processes are complete and the chip is dry, can usually be "swept off" by the air-brush heads. Sometimes more than one sweep across the chip is required.

Because the unusual chip motions are not convenient, the extra sweep motion is incorporated with the chip-reposition motion. Instead of merely moving a short distance forward or backward, the chip is moved a full two frames (in that direction where there is enough room on the chip) and then returned to within a short distance from where it started (according to the repositioning requirement). This movement is appropriate, for it is as reasonable to assume that an extra sweep is needed as it is to assume a repositioning is needed, if a line cannot be corrected by other techniques.

Recovery schedule

Table 1 shows a list of problems and a list of tools relevant to the problems. How should these tools be exploited? Actually, very little is known about the nature of a given problem line, and often that which can be known is not worth the processor time needed to find it out. Alternatively, a statistical approach was used to design a schedule of correction attempts using different tool combinations.

The initial objective of the schedule is to correct all correctable lines in any block of data in minimum time. This implies that, for any given attempt to correct a line, the tool combination to use is the one which statistically yields the most frequent success for the least amount of time invested. This philosophy leads to a list of actions to be taken against an offensive line. If all actions on the list are exhausted and the line is still not corrected, the line is considered unreadable.

The recovery schedule for normal READ is shown in Table 2. It is a program of actions to be taken to correct a problem line. Each action is a series of READ attempts operating under a particular combination of recovery tools specified for the action in Table 2. Also specified for each action is an EDC schedule which specifies what levels of EDC will apply to each of the READ attempts for the action. Table 3 shows a list of available EDC schedules. For example, schedule B_2 specifies the first READ attempt using no EDC at all, then 4 attempts using 2-EDC, then 3 using 3-EDC, then 1 with 4-EDC, and finally 1 with 5-EDC. If by this time the line is not corrected, the controller goes to the next action on Table 2, which may specify a different set of tools and EDC schedule.

If the reader dwells on a line and indicates "reader cycling," the current action is abandoned. The next action is entered, but the index to the EDC series is not reset. This assures that higher levels of EDC will be attempted.

The first actions attempted are for speed. Later actions are oriented to more effective recovery, but at an increasing cost in system time. It is a rare event to get past action 12 in the strategy (about 0.04% of all lines). However, when this occurs, it is vital to do everything possible to find, start and read the line.

Remaining are two more tools: mechanical chip-repositioning and chip sweeping, which have been combined into one mechanical motion. About 250 milliseconds are

Table 2 Normal READ action schedule (90-second timeout).

Action number	*Forced ALS*	*Extended coasting*	*Hard sync*	*Offset ride high*	*Offset ride low*	*Line jump*	*Re-*READ *with normal EDC schedule*
00							A*
10							A*
20							B
30							B
40							B
50						x	B
60						x	B
70						x	B
80					x	x	B
90					x	x	B
100					x	x	B
110					x	x	B
120			x		x	x	B
130			x		x	x	B
140			x		x	x	B
150			x		x	x	B
160		x			x	x	C
170		x			x	x	C
180		x			x	x	C
190		x			x	x	C
200		x	x	x		x	C
210		x	x	x		x	C
220		x	x	x		x	C
230		x	x	x		x	C
240		x	x		x	x	C
250		x	x		x	x	C
260		x	x		x	x	C
270		x	x		x	x	C
280	x		x		x	x	C
290	x		x		x	x	C
300	x		x		x	x	C
310	x		x		x	x	C
320	x				x	x	C
330	x				x	x	C
340	x				x	x	C
350	x				x	x	C
360		Do chip reposition and sweep, and repeat actions 00–350. After 7 cycles, go to 370.					
370						x	C

* A on first recovery cycle entry, C on subsequent cycles.

invested in the mechanical motion. When the motion is complete, the situation on the chip may be entirely different. A flaw may have been crossed over or swept away. Consequently, the next technique is a complete reapplication of all nonmechanical recovery procedures. This has been called a data-recovery cycle. There are nine such cycles separated by eight mechanical motions in the over-all recovery schedule for reading. A reposition in the forward direction may place the desired line out of the window, but the second motion (in the backward direction) can compensate for the first motion, still with the desired effect of crossing over a flaw. One reason for eight repositionings is to compensate rare but existent mispositionings of the chip.

• *Rereading*

The reason for the effectiveness of multiple tries is that each READ attempt is statistically different. The data detector is designed so that, if data bit waveforms are garbled, there is about an even chance of producing a "1" or "0" bit; also, electrical noise and mechanical vibrations are different for each reading, and so on. Re-READ's are also statistically different because various recovery tools produce different effects. For example, a problem line is approached from both of the adjacent lines, and the offset scan causes the beam to penetrate flaws at different vertical positions to produce a variation in the data signal disturbances, etc.

For data recovery each combination is repeated several

Table 3 EDC schedules.

Schedule		*Attempt number* 0	1	2	3	4	5	6	7	8	9
Normal read schedules	A_1	0	1	1	1	1	1	—	—	—	—
	B_1	0	2	2	2	2	3	3	3	4	5
	C_1	0	4	4	4	4	4	4	5	5	5
Readback-check schedules	A_2	0	1	1	1	1	1	—	—	—	—
	B_2	0	1	1	1	2	2	2	3	3	3
	C_2	0	3	3	3	3	3	3	3	3	3
Auxiliary line-start	C_1	0	3	3	3	3	3	3	3	3	3

These entries indicate the maximum number of errors which can be corrected in the respective attempt. Higher maximums imply longer correction times.

times for re-READ effects. A line may be reread more than 2,400 times before it is considered unreadable. This takes about 13 seconds.

If the recording medium were even slightly subject to wear in the process of rereading, these data recovery procedures would be prohibited. Even in areas where there is a flaw but no unreadable lines, there is some recovery activity for each of several lines, adding up to a large number of re-READ's in that vicinity.

• *ALS in the schedule*

Table 2 shows that a forced auxiliary line-start is scheduled late in the recovery cycle. This is so because false line-starts are rare, and it is never known for sure when a false line-start is the problem. Further, ALS is by its nature successful in only 10 to 30% of its attempts, and thus there is a lower probability of success for the technique using ALS.

In contrast, nonline-start can be detected by hardware. It is probable that the scanning spot is on the correct line, so that immediate use of ALS is warranted and carried out automatically in hardware. Whenever ALS is used, be it forced or not, 3-EDC is used to correct the line, regardless of the EDC schedule currently in use.

Readback-check considerations

In this mass storage system, writing is always done on new material. A large quantity of recording medium is then necessarily subject to high standards of quality control. Still, there are flaws in "acceptable" bulk recording film. Some of these flaws prove to be unacceptable after recording and development. To eliminate the unacceptable flaws, a readback-check (RBC) operation is performed shortly after recording.

For reading in RBC mode, the data-recovery activity is curtailed somewhat to be conservative. A chip is rejected if one line in it cannot be read in RBC mode. The rewriting penalizes throughput and requires additional data buffering.

Of primary interest here is the diminished data-recovery schedule for the RBC mode. The schedule is shorter, and some data-recovery features are not used. In particular:

(1) Correction for 4 and 5 errors is avoided. (This is done because one additional small 2-bit flaw can cause two marginal characters, one marginal bit in each character.)
(2) The number of re-READ's is reduced.
(3) Picker[5] repositioning is used six times instead of eight.
(4) EDC schedule C, forced ALS, and parameter variations are not used.

The READBACK check schedule is shown in Table 4.

System performance

The control processor, in addition to running the mass-storage system, gathers statistics on its performance. One portion of these statistics relates exclusively to data recovery and unreadable lines.

Tables 5 and 6 are statistical printouts for normal READ and READBACK checks, respectively. Each row in the matrix corresponds to one action in the schedule. The columns correspond to successive EDC attempts. The schedule of actions starts on the top row and the schedule of EDC attempts on the first column.

To illustrate an example, there is a "12" printed at the intersection of row 9 and column 2 on Table 5. Row 0090 corresponds to Action 9, which is a direct jump with low off-set scanning and applying EDC schedule B. Column 2 corresponds to the second attempt at 2-EDC in EDC schedule B, as seen from Table 2. The number "12" indicates that 12 lines were corrected by this combination of recovery tools during the period of time over which these statistics were accumulated.

The first attempt at line jump usually represents the arbitrary jump and the second in the sequence is the direct jump; the second is therefore more successful.

Unfortunately, all 9 data recovery cycles (separated by picker repositioning) are accumulated in this one table. This causes a few entries in rows 1 and 2, columns 6 through 9. It also shows an accumulation of successes for forced auxiliary line start in rows 280 through 310; actually, these are spread out through all 9 data recovery cycles. Separate statistical printouts indicated the approximate success of picker repositioning and auxiliary line start.

The last column in Tables 5 and 6 (percent/line) has the computed percentage of successes for each row

Table 4 READBACK check (30-second timeout) action schedule.

Action number	*Forced ALS*	*Extended coasting*	*Hard sync*	*Offset ride high*	*Offset ride low*	*Line jump*	*Re-*READ *with* READBACK *check EDC schedule*
00							A
10							A
20							B
30							B
40							B
50						x	B
60						x	B
70						x	B
80						x	B
90						x	C
100						x	C
110						x	C
120						x	C

NOTES: (1) Prior to each action, forced looping is reset.
(2) Regions of bad data can cause the time to process one column of frames on the chip to be excessive. A 90-second timeout for READ, and a 30-second timeout for READBACK check cause further efforts for that column to be aborted.

Table 5 Normal READ (29,632 corrections for about 4 × 10^6 lines); see text.

ACCUMULATION OF SUCCESSFUL TRY COUNTS

	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	PERCENT/LINE
		23210	1937	656	346	198	31	34	28	31	.905953405
0010	89	180	143	93	86	70	26	35	22	21	.02276039
0020	32	67	622	309	94	220	76	74	105	75	.049805118
0030	12	3	24	14	15	17	18	9	19	32	.004849595
0040	3	1	10	10	2	8	7	10	15	20	.00255868
0050	.	.	.	1	.	1	.	.	4	1	.000208260
0060	.	.	1	3	2	4	2	1	3	3	.000565288
0070	.	2	4	.	4	7	4	2	3	2	.000833055
0080	.	1	3	3	3	7	8	4	4	4	.001100824
0090	.	3	12	6	5	5	5	3	10	2	.001517354
0100	.	2	4	9	8	3	4	5	4	7	.001368595
0110	.	1	4	1	1	9	2	1	2	3	.000714044
0120	.	1	1	5	2	5	.	2	3	5	.000714044
0130	.	.	2	.	7	1	2	2	.	2	.000476029
0140	.	.	3	1	3	1	4	6	1	.	.000565288
0150	.	2	.	2	2	2	1	2	5	4	.000595036
0160	.	1	2	1	.	2	3	2	.	2	.000386771
0170	.	.	2	.	.	1	.	.	1	1	.000148753
0180	.	1	1	2	2	3	.	3	2	2	.000476029
0190	.	.	2	.	2	3	2	1	3	.	.000386771
0200	.	1	1	.	2	2	.	.	.	.	.000178506
0210	.	.	.	.	.	.	2	.	.	.	.000059499
0220	.	.	.	.	.	.	2	.	.	1	.000089249
0230	.	.	.	1	.	.	2	.	1	.	.000119003
0240	.	.	.	.	.	.	.	1	1	2	.000119003
0250	.	3	1	1	1	.	.	.	.	1	.000208260
0260	.	.	2	2	.	2	1	1	1	.	.000267763
0270	.	1	1	.	1	2	1	.	.	.	.000178506
0280	1	1	9	7	2	3	2	2	1	1	.000862807
0290	2	2	1	1	2	3	3	4	.	3	.000624790
0300	1	.	3	2	1	1	2	3	2	.	.000446276
0310	1	.	4	4	2	.	2	2	2	.	.000505781
0320	.	.	.	.	.	2	.	.	1	.	.000089249
0330	.	.	.	1	.	.	1	.	.	1	.000089249
0340	.	.	.	.	.	1	1	.	.	1	.000089249
0350	.	.	1	.	.	.	.	.	.	.	.000029747
0370	.	.	.	.	1	.	.	.	.	1	.000059499

Table 6 READBACK check (15,201 corrections for about 2×10^6 lines).

ACCUMULATION OF SUCCESSFUL TRY COUNTS

	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	PERCENT/LINE
		12052	1120	419	214	148	10	22	19	7	.931906609
0010	65	139	108	49	52	39	6	10	10	5	.027637898
0020	24	39	3	3	1	3	2	3	4	4	.004921031
0030	13	30	3	3	8	3	5	4	2	2	.004177153
0040	17	32	4	1	1	1	3	.	1	1	.003490496
0050	1	3	2	2	1	2	.	.	.	.	.000629431
0060	2	6	6	7	1	1	4	2	1	1	.001773856
0070	2	.	.	2	1	.	.	2	.	1	.000457766
0080	1	.	.	.	1	2	.	.	.	.	.000228881
0090	.	1	30	8	2	2	2	1	.	.	.002632177
0100	.	2	42	31	6	8	4	1	2	.	.005493246
0110	.	1	68	57	23	13	6	10	4	1	.010471498
0120	.	2	19	25	31	7	10	3	3	8	.006179899
0370	.	.	.	.	.	.	.	.			

(action) in that run. The percentage is based on the total number of corrections accomplished in the run.

The very last action is a line jump to the offending line. The uncorrected line is stored by the controller with other unreadable lines (if not corrected by this last try) for off-line manual analysis.

Conclusions

To our knowledge, this is the first practical implementation of a very complex error detection and correction facility with such powerful capabilities. It increases greatly the system's tolerance of disturbances to the recorded data but requires support functions that provide equal tolerance for data synchronization and identification. The speed of error correction was found to be an important consideration which has not received much theoretical attention. A statistically optimized schedule for the application of recovery tools is fundamental to full exploitation of the recovery capabilities. It is believed these procedures will also enhance the long-term archival quality of the silver film medium.

Acknowledgments

Significant contributions to this effort were made by the following persons: R. A. Jensen and H. W. Johnson, photo-digital reader; J. H. Davis and T. E. Tate, data controller; B. D. Cunningham, programming; and I. B. Oldham, coding theory.

Footnotes and references

1. I. B. Oldham, R. T. Chien, and D. T. Tang, "Error Detection and Correction in a Photo-Digital Storage System," *IBM J. Res. Develop.*, **12,** 422–430 (1968).
2. An operation in modulo 2 arithmetic. See W. W. Peterson, *Error Correcting Codes,* MIT Press, Cambridge, 1965.
3. The probability of an undetected error is very slight—less than 1 in 10^{10} lines. Consequently, the absolute terms "errorless" and "corrected" are common usage.
4. After six wrong results have occurred, the program will execute some other search procedures, as indicated in the section entitled "Coarse compare."
5. The "picker" is the device that extracts the film chip from the cell.

Bibliography

Bartee, T. C., and D. I. Schneider, "An Electronic Decoder for Bose-Chaudhuri-Hocquenghem Error Correcting Codes," *IRE Trans. on Information Theory,* **IT-8,** S17 (1962).

Bose, R. C., and C. R. Chaudhuri, "A Class of Error-Correcting Binary Group Codes," *Information and Control* **3,** 68 (1960).

Chien, R. T., D. T. Tang, E. S. Barrekette, and A. M. Katcher, "Analysis and Improvement of Photostore Error Rates," *Proc. IEEE* **56,** 805 (1968).

Furman, R. M., "IBM Photo-Digital Mass System." *IBM San Jose SDD Technical Report* 02.427 (1968).

Gorenstein, D., and N. Zierler, "A Class of Error-Correcting Codes in p Symbols," *Journal of the Society for Industrial and Applied Mathematics* **9,** 207 (1961).

Gustlin, D. P., and D. D. Prentice, "Dynamic Recovery Techniques Guarantee System Reliability," presented at the 1968 Fall Joint Computer Conference.

Haughton, K. E., "An Electron Beam Digital Recorder," *Third International Electron and Ion Beams in Science and Technology Conference,* 1968.

Hocquenghem, A., "Codes Correcteurs d'Erreurs," *Chiffres* **2,** 147 (1959).

Kuehler, J. D., and H. R. Kerby, "A Photo-Digital Mass Storage System," *Proceedings of the Fall Joint Computer Conference,* p. 753, 1966.

Peterson, W. W., *Error-Correcting Codes,* MIT Press, Cambridge, Massachusetts, 1961.

Reed, I. S., and G. Solomon, "Polynomial Codes Over Certain Finite Fields," *Journal of the Society for Industrial and Applied Mathematics* **8,** 300 (1960).

Received July 30, 1968