DATA STRUCTURE FOR CODING SCREENED HALFTONE PICTURES

Søren Forchhammer*
Morten Forchhammer*

Abstract:

A method of coding scanned documents containing halftone pictures, e.g. newspapers and magazines, for transmission purposes in prepress is proposed. The halftone screen is estimated and the grey value of each dot is found, thus giving a compact description. At the receiver the picture is re-screened. A new data structure and related algorithms for handling the digital screen without restrictions on the screen parameters is presented. Data compression rates above 20 are obtained for halftone pictures. The algorithms are suited for implementation using fast dedicated hardware. The rescreening can also be used as digital halftoning with arbitrary screens using look-up tables.

1. Introduction

Line art with text, graphics and screened halftone pictures may be scanned and transmitted in the prepress process of newspaper and magazine production. Presently, often the CCITT Facsimile Group IV MMR (modified Modified R E A D) code (CCITT, 1984) is being used for data compression. The code is optimized to compress scanned documents with text and graphics. It does not perform well on screened halftones. Usubuchi et al. (1980) have implemented an adaptive predictive runlength code for newspapers with halftone screen at 45°. It achieved a compression rate of 5. Crossfield has implemented a similar method (Jordan, 1985). The method utilizes both the redundancies of the grey value images and the redundancy of halftone coding of the image. Though the redundancies are not fully utilized. As the halftone images generate most of the bits in this case higher compression rates are desirable. Chao (1982) presented a two-channel scheme. First the halftone is converted to a continuous tone image which is transmitted in one channel. The error image of the rescreened compared with the original halftone is found and transmitted in the other channel. This way a compression rate of about 10 is obtained.

The method operates directly on a computer generated halftone image, and presumes knowledge of the exact screen structure and phase.

* Institute of Telecommunication, Technical University of Denmark

The principle of Chaos first channel is here proposed generalized to handle scanned (distorted) screens without prior knowledge of the screen, and instead estimating the screen structure and phase blockwise locally.

It is proposed to segment the document in text/graphics and screened halftones and code the segments with different codes. The proposed method involves segmenting the page, estimating screen parameters, converting the halftone to contone, and at the receiver re-screening the image. The data structure presented also has the advantage of taking care of some of the effect of the second channel of Chaos scheme.

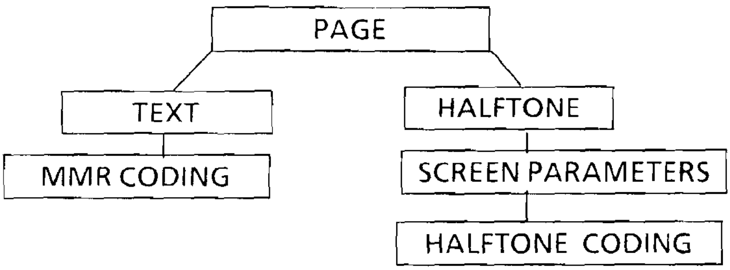

Figure 1. Block diagram.

The re-screening process is similar to the process of digital halftoning. Digital halftoning with arbitrary screens has been a problem of the graphic industries. The problem is to handle arbitrary screen rulings and angles often with irrational parameters in the inherently integral setting of digital processing.

The general format of the coded data is shortly described in Section 2. An overview of some methods using descreening to code halftone images are given in Sections 3 and 4, with emphasis on the handling of the screen. Approximate compression rates for these methods are given in Section 5. In Section 6 results of one implementation within the general coding format are presented.

2. Data format and segmentation

The pages are scanned at high resolution creating a binary image. The line art pages are treated alike whether they have color attributes or not connected to them. A page is divided in blocks of say 256 by 256 pixels. Each block is segmented to be a text or a halftone block. Each block has a header telling the type of the block. For halftone blocks the necessary screen parameters are also in the header.

The segmentation may be performed by coding each block using the facsimile MMR code. Blocks resulting in more than some threshold value of coded bits become potential picture blocks. Blocks with its two preceding direct neighbors being (potential) picture blocks become picture blocks. The picture blocks are tried coded as a halftone block. If this is not successful the blocks are coded as text with the MMR code. Within halftone blocks the grey values of the image are encoded, with possibilities for special edge preservation as described later.

3. Describing the halftone screens

3.1.1. Digital grids

Generally, the halftoning screen before scanning can be described in a continuous coordinate system (r,c) with integer values at the screen dot centers. The scanner points can be described in another continuous coordinate system (x,y) with integer values at the scanner pixel centers. There is a bijective relation between the two coordinate systems. This relation is the same as the relation of deformation models employed in remote sensing (Niblack, 1985).

$(r(x,y);c(x,y)),(x,y) \in R^2$ describes the screen coordinate system. $(x(r,c);y(r,c)),(r,c)\in Z^2$ gives the black screen dot centers and $(x(r+\frac{1}{2},c+\frac{1}{2});y(r+\frac{1}{2},c+\frac{1}{2})),(r,c)\in Z^2$ gives the white screen dot centers. Rounding of the center values gives digital grid points. Let the prescript digital denote a digitized representation in the scanners coordinate system.

For a linear grid the partial derivatives of r and c with respect to x and y are constant. The centers are given by

$$x(r,c) = cV_{1x} - rV_{2x} + e_x \tag{1}$$

$$y(r,c) = -cV_{1y} - rV_{2y} + e_y \tag{2}$$

where $(x,y) \in R^2$, $(e_x,e_y)\in[-\frac{1}{2},\frac{1}{2}[^2$, $(r,c)\in Z^2$ and $(V_{1x},V_{1y},V_{2x},V_{2y})\in R^4$

In scanned line art the (r,c) coordinate system of the halftoning screen may vary across the (x,y) coordinate system. A varying description may therefore be desired. In a N by N pixel block with (r,c) values with subscripts a,b,c and d at the corners (Figure 3) controlling the grid, the (r,c) coordinate system may be described by a polynomial description, giving an indirect description of the centerpoints' (x,y) coordinates.

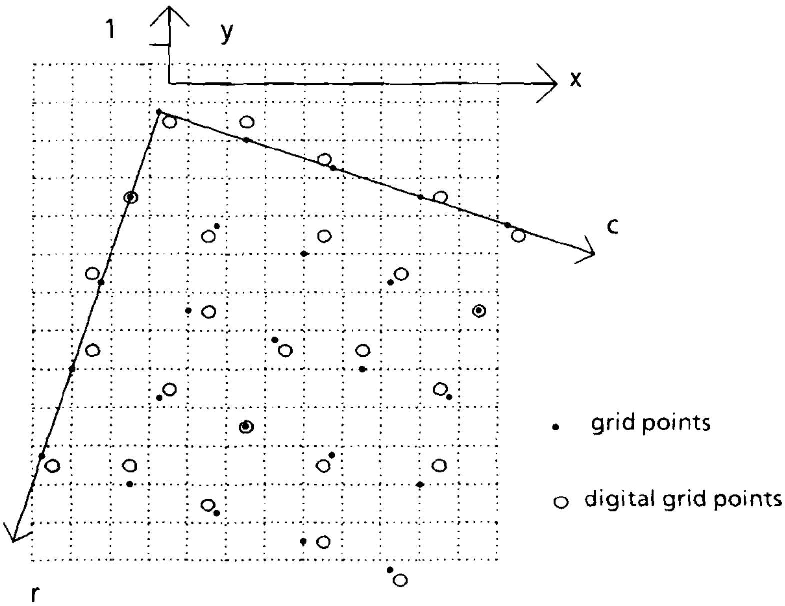

Figure 2. Digital grid points.

$$\begin{pmatrix} r \\ c \end{pmatrix} = \begin{pmatrix} r \\ c \end{pmatrix}_a + \left[\begin{pmatrix} r \\ c \end{pmatrix}_b - \begin{pmatrix} r \\ c \end{pmatrix}_a \right] \frac{x}{N} + \left[\begin{pmatrix} r \\ c \end{pmatrix}_c - \begin{pmatrix} r \\ c \end{pmatrix}_a \right] \frac{y}{N} +$$

$$\left[\begin{pmatrix} r \\ c \end{pmatrix}_d - \begin{pmatrix} r \\ c \end{pmatrix}_c - \begin{pmatrix} r \\ c \end{pmatrix}_b + \begin{pmatrix} r \\ c \end{pmatrix}_a \right] \frac{xy}{N^2} \tag{3}$$

With a polynomial description of the coordinate system the equation corresponding to (1, 2) yields the exact differential form

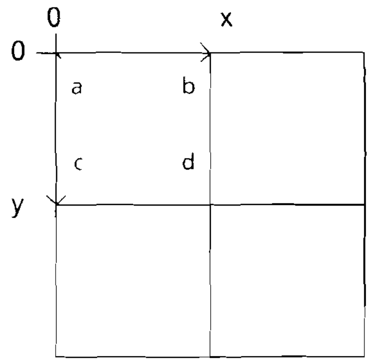

Figure 3. The notation at the corners of a block

$$x(r,c) = \int -V_{1x}\delta c - V_{2x}\delta r$$

$$y(r,c) = \int V_{1x}\delta c - V_{2x}\delta r$$

Where

$$V_{1x} = \frac{K_{ry}}{K_{cx}K_{ry} - K_{rx}K_{cy}},\; V_{1y} = \frac{-K_{rx}}{K_{cx}K_{ry} - K_{rx}K_{cy}}$$

$$V_{2x} = \frac{-K_{cy}}{K_{cx}K_{ry} - K_{rx}K_{cy}},\; V_{2y} = \frac{K_{cx}}{K_{cx}K_{ry} - K_{rx}K_{cy}}$$

$$where\, K_{ry} = \frac{\delta r}{\delta y}, K_{rx} = \frac{\delta r}{\delta x}, K_{cy} = \frac{\delta c}{\delta y}, K_{cx} = \frac{\delta c}{\delta x}$$

The linear screen is faster to generate as it may directly be generated incrementally using a straight line algorithm for each of the coordinates. The polynomial screen gives more flexibility

3.2. Data structure

Given the digital grid points the bit plane is uniquely tesselated by drawing digital straight line segments (DSLS) between the digital grid points which are direct neighbors. The lines can be drawn incrementally by Bresenhams algorithm (1965). Drawing lines between the digital black centers gives a discrete approximation of white screen cells and vice versa.

The approximating screen cells are not a digitization of the lines bounding the screen cells nor do they have the uniformity in area usually required (variations of the area is limited to one) in digital halftoning. These problems are circumvented by using white screen cells in dominantly dark areas and vice versa. In this way the exact position of the screen cell boundaries is not critical. The structure has the advantage of being easier to handle than other structures (see Section 3.4). The structure implies that the type of screen cells may change.

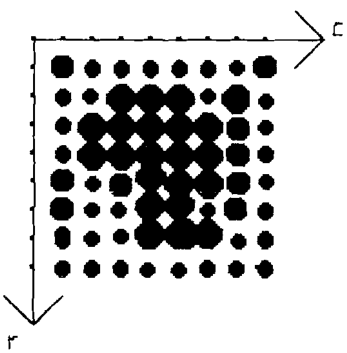

Figure 4. Bit map

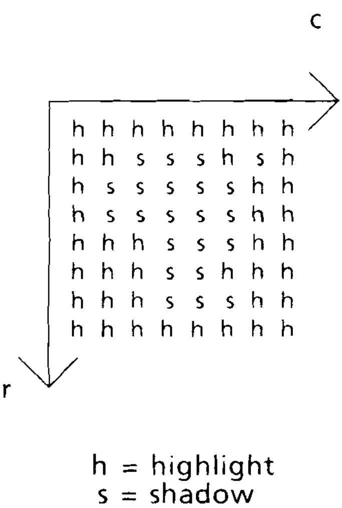

Figure 5. State of black cells

Drawing DSLS from the centers of a screen cell to the four corners will partition the screen cells into four digital triangles. When changing type of screen cells there will be a brim af alternating black and white partial screen cell triangles (see Figure 6). Given the grey values in the digital black (or white) screen cells it is possible to segment the picture into shadow and highlight areas.

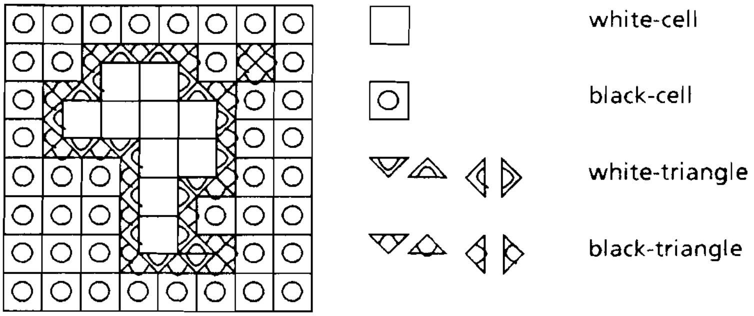

Figure 6. Segmenting the image of Figure 4 in cells and triangles

The grey values are thresholded into three groups 1, 2 and 3 denoting light, middle, and dark grey values. The middle values may be grouped with the highlight or the shadow cells in any desired way according to local properties. This can be done traversing the screen cells in some order using hysteresis. Highlight cells constitute the light area. The shadow area is white screen cells with four dark cells at the corners. Other combinations belong to the border. At the border, black cell triangles are set at shadow points facing a highlight point and white cell triangles are set if the long triangle side connects two shadow points (Figure 7), (Forchhammer, 1987a), (Forchhammer, 1988).

This theme gives an implicit edge detection, for edges crossing the hysteresis interval. At these edges a quarter of a screen cell is coded at a time. This complies with the advice of using a resolution of the continuous image, which gives four grey-values per screen dot, (Kekolahti, 1982). More elaborate edge detection may be applied. Generally, the sender (and receiver) may employ any desired processing of the grey values, e.g. unsharp masking techniques.

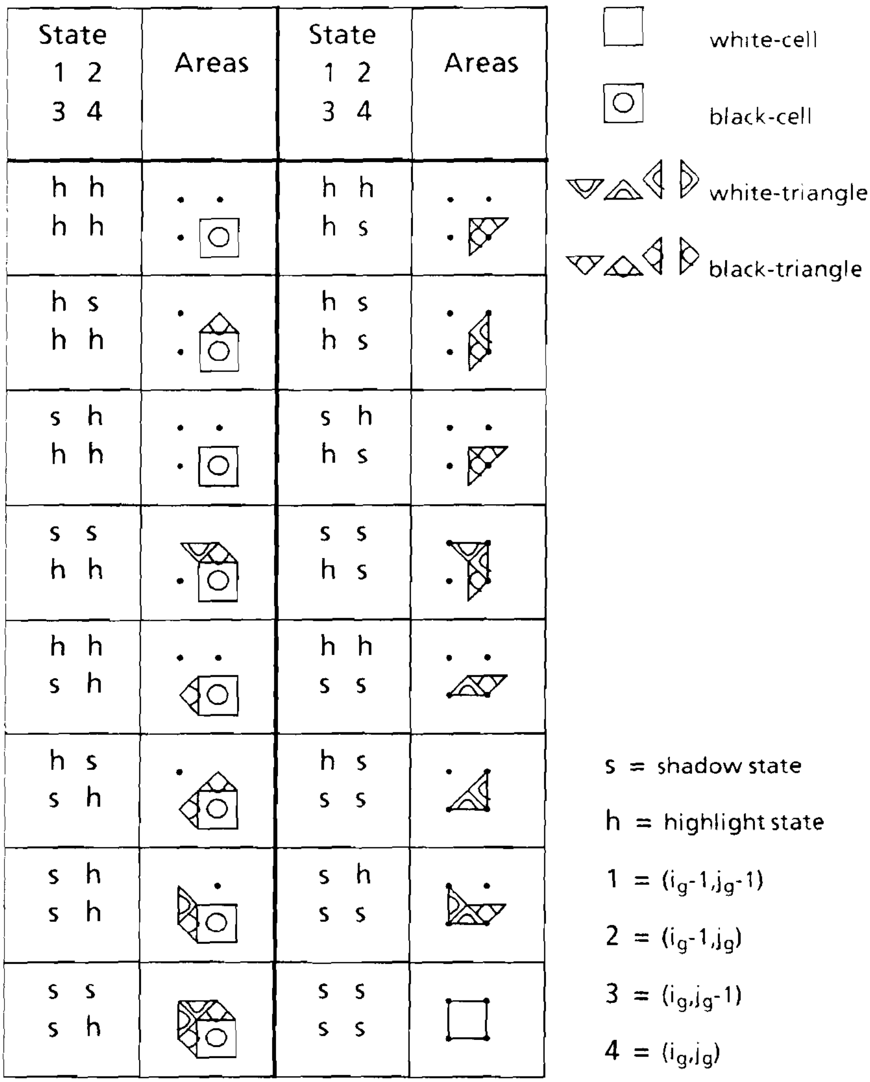

Figure 7 State diagram segmenting the image in cells and triangles

3.3. Estimating screen parameters

It is necessary to estimate the screen parameters from the halftone image when using the data structure above.

Wilson (1986) finds the centers of the screen dots and uses least squares fitting to estimate the grid parameters. This method is very time consuming.

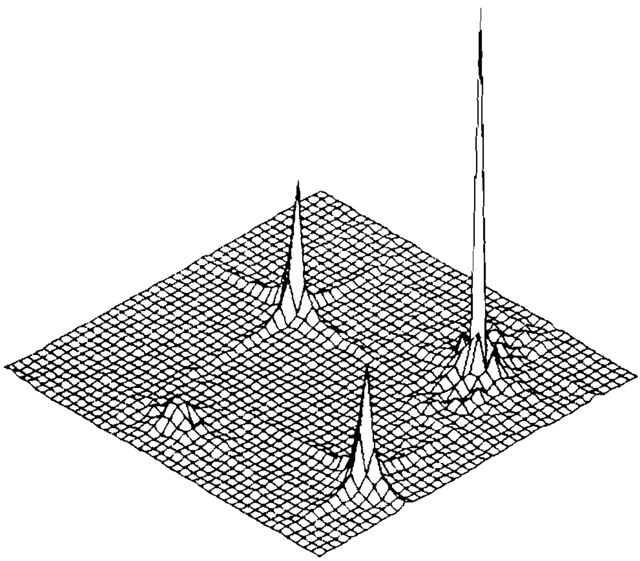

Figure 8. Lower part of spectrum of a halftone picture.

The screen parameters may be estimated using the Fourier transform. Allebach (1970) has derived an analytical expression of the Fourier spectrum of digital halftone pictures. It shows that there are peaks at the frequency corresponding to the interdot spacing and the screen angle (Figure 8). The calculation of the Fourier transform is also quite time consuming. Finding approximate positions of centers by some simple preprocessing the same result as using the Fourier transform may be obtained at less computational cost, with a Hough transform based method (Baird and Forchhammer, 1988). The method is more robust than least squares fitting. It may be used to find a near accurate estimate and in indentifying a block as a halftone block. The method also has the advantage that if the (feasible) range(s) of parameters are known it can be executed faster.

Having a near accurate estimate it is possible to use the four digital triangles (or squares) of each screen cell. The grey values of the triangles should balance all four of them or pairwise depending on the dot structure. The digital points with locally best balance are digital grid points. When the approximate angle is known it is enough in each picture block to estimate $(r,c)_d$, from finding a center of a dot in the lower right corner, to calculate the grid of that block using (3). If descreening the halftone image with triangles the balance of the points can also be used to check if the approximate grid is accurate enough.

3.4. Analysis of digital lattices

Coding the halftone images involves estimating and coding the screen grid. It is a question of how many bits are used for coding the screen and how many different shapes of digital cells and triangles are necessary for masking out the grey values.

Making efficient use of the redundancy and obtaining algorithms with low complexity are the main goals, which sometimes contradict each other.

Basically, there are the possibilities of operating with the description of the whole grid, obtaining the cell boundaries by digitizing e.g. (1) and (2) or (3) or base the operations on the grid points, i.e. for integer (r,c). If grid points are used they can be coded for the whole block or grid point by grid point relative to the former grid point.

Coding the grid points blockwise it can be shown that having a digital grid of order n, i.e. $(r,c)\in[0,n]^2$ there exists a rational approximation generating the same digital grid points with approximations

$$(V_{1x},V_{2x}) \approx \left(\frac{p_{1x}}{q_x}, \frac{p_{2x}}{q_x} \right), (V_{1y},V_{2y}) \approx \left(\frac{p_{1y}}{q_y}, \frac{p_{2y}}{q_y} \right)$$

with $q \leq 2n^2$, (Forchhammer, 1988)

This shows that the number of different digital grid point sets are $O(n^{16})$. If $V_{1x} = V_{2y}$ and $V_{1y} = V_{2x}$ the number of grid point sets are $O(n^{10})$.

For a linear grid with (known) real valued parameters,there are at most n^4 different digital sets of points when digitizing a given set of $n \cdot n$ points of a constant grid given a constant grid basis. It may be proven by translating the area of digitization of one point of the set to all the others and observe the bisection parallel to the x and y direction, respectively. Every point cuts the translated square at most once.

Coding the grid points relative to the former grid point is equivalent to coding the vector between them. For a vector without integral coordinates, then digitizing the two points and defining a digital vector by the digital points gives any one of four different vectors. Generalizing this gives an upper limit of $4^{(n^2-1)}$ different digital sets of points when digitizing n^2 points with one reference point. For a triangle the upper limit is 16 and for a square 64.

Given a digital cell, let N be the length of the longest of the projections on the x- and y-axis, respectively. There are $4N^2 + O(N)$ different digital shapes. (Forchhammer, 1988). With $N = 12$ this gives approximately 600 possibilities.

The gain in quality of handling all the different shapes caused by digitization is marginal. Therefore an approximating screen is used. The approximating screen cells are defined by the digital grid points at the corners. This gives upper bounds of 64 shapes for the cells and 16 shapes for the triangles. The grid points are coded for the whole block using significantly less bits than coding the grid points relative to a neighbor. The number of bits used for coding the grid for a whole block is more or less negligible, so the above choice is based on complexity considerations.

4. Determining the grey values

The grey values can be found in one of two ways:

1) Counting the number of pixels inside squares with the approximate size of the screen cell or a multiple thereof. This method is used in offset to gravure conversion (George and Toor, 1983). The method implies a low pass filtering.

In one method of rescreening, the receiver uses the data structure described previously, and the grey value associated with each screen cell is determined by the grey values of the four nearest centers. The grey values are weighted proportional to the distance squared from the opposite center to the center in question (Stucki, 1979), (Forchhammer, 1987a). This is the same as using the data structure for digital halftoning with arbitrary screen using look-up tables. This is a bilinear interpolation and uses an indirect transformation, (Niblack, 1985).

2) Following the screen structure the grey values of the screen cells are found. This preserves the contours of the images to a higher degree than the first method.

The first method is straightforward to implement. The second method implies use of the data structure described previously. Using triangles 4·16 different triangle masks are used at the sender to mask the screen dot triangles and 64 triangles at each grey level at the receiver

Another parameter of the screened halftones is the dot shape. The shape of the generally used screen dots: circles, squares, ellipses and diamonds may easily be described analytically. The circle-square dot is described analytically in (Stucki, 1979). This gives a size independent description which the user may use to regenerate the dots, given the type of shape they have. An extension would be to build a dot shape library from the data. This is similar to coding by pattern matching, (Johnsen et al., 1983).

5. Data compression

A document with a_t of it coded using the MMR code and a_h of it coded as halftone may be viewed as a composite source with compression ratio R

$$R = a_t R_t + a_h R_h$$

where R_t, R_h are the compression ratios of text and halftone blocks, respectively. The compression ratios are the inverse figures of the compression rates.

Coding the screened halftones by their grey values gives a compression ratio of approximately

$$R_h = \frac{\lceil log_2 n \rceil}{n} \quad (4)$$

where n is the number of pixels per screen cell.

The compression rate of text of a given size using the MMR code is proportional to the resolution in the range 200-400 pels/inch, (Bodson, 1985). A result that probably may be extrapolated to higher resolution because coding each contour pixel requires the same number of bits, regardless of the resolution, (Schreiber, 1986) and going to higher resolutions the contours dominate. This gives an empirical formula:

$$R_t \approx \frac{100}{3.5r} \quad (5)$$

where r is the resolution in pels/inch. The result of the descreening of Section 3.2. can be coded as follows block by block. Each block has a header with the grid coordinates of 1-4 of the block corners. The black screen cells are traversed back and forth row by row. The state of each black cell is determined, if it is different from the former cell a change sign is transmitted. The grey values of the cells and triangles are transmitted according to Figure 7. The grey value of each cell/triangle has a specific interval within which the value is coded. A formal description is given in (Forchhammer, 1988). At the borders of the blocks, the cells/triangles overlap two blocks. These cells/triangles have to be coded twice if the blocks are coded and decoded independently of the grey values of the neighboring blocks.

6. Results

The four 256·256 blocks of the 512·512 test picture of Figure 10 were coded as described above. The same ordering of a-d as in Figure 3 is used. The results are given in Figure 9.

Block	Coding, storing border values	Coding blocks independently	Coding grey values
a	22.6	20.1	21.2
b	20.5	18.3	21.2
c	24.4	21.5	21.3
d	23.8	21.3	21.7
512·512	22.7	20.2	21.3

Figure 9. Compression rates. Column 2 and 3 gives result of a code based on Figure 7. Column 4 gives the result of coding the grey values.

The halftone in Figure 10 is scanned at approximately 800 lines/cm (2000 l/inch) and the screen ruling is approximately 60 lines/cm (150 l/inch). Descreening the halftone and coding the grey values would give a data compression of 22.2. Comparing with simple grey value coding, coding the change and the triangles when changing requires more bits, coding the cells, here as in most cases, requires one less bit. All in all for the test picture, coding changing between black and white cells gives a little higher data compression.

Viewing four color separations under one and assuming; the text is 80% of the final document but only on the black separation, i.e. $a_t = 0.2$; R_t is given by (5); 20% is screened halftone pictures and coded as such, i.e. $a_h = 0.2$; $1/R_h \approx 22$, an average value of Figure 9; the scanner resolution is $r = 2000$ lines/inch; the data compression becomes

$$R_h \approx 0.045, \quad 1/R_h \approx 22$$

$$R_t \approx 0.014, \quad 1/R_t \approx 70$$

$$R \approx 0.012, \quad 1/R \approx 84$$

Figure 10 Binary image of the original 512 by 512 scanned halftone picture.

The value on 1/R is an upper bound on the compression rate, as the result presumes coding all halftone areas, as halftones. Failing to detect areas with halftones, and coding them with the MMR code will reduce the compression rate. The compression may on the other hand be increased by employing conventional image coding, as predictive or transform coding to the grey values of the image.

In Figure 10 the orignal binary picture is shown after scanning and thresholding but before compression. In Figure 11, 12, and 13 the picture is rescreened using the approximating grid and cells and triangles. In Figure 11 and 12 determination of the grey values by the first method has been used. Before rescreening bilinear interpolation has been applied. The effect of the low pass filtering is seen in areas with larger contrast.

Treating each halftone block separately using a linear grid may give visually disturbing results, if the original grid is varying, when putting

Figure 11. Reconstruction in four blocks using linear grid.

neighboring blocks together (Figure 11). Using the polynomial grid description of Section 3.1 secures 'phase continuity' across the block borders giving a satisfactory visual result (Figure 12 and Figure 13). Also when putting a block of the original bit map together with blocks re-screened with a polynomial grid, the result is satisfactory (Figure 12). (The upper right block is the original bit map). This might occur if a block with (part) halftone is coded using MMR.

In Figure 13 the descreening was done following an estimate of the grid in the original. As expected, this gives a better preservation of the edges. Figure 13 was reconstructed using triangles in the 'middle grey value area' and not just between highlight and shadow area. This gives potentially a higher resolution in the 'middle grey value area' compared with the method of Figure 7. On the other hand the distortion of the shape of the dots would be reduced in this area with the method of Figure 7.

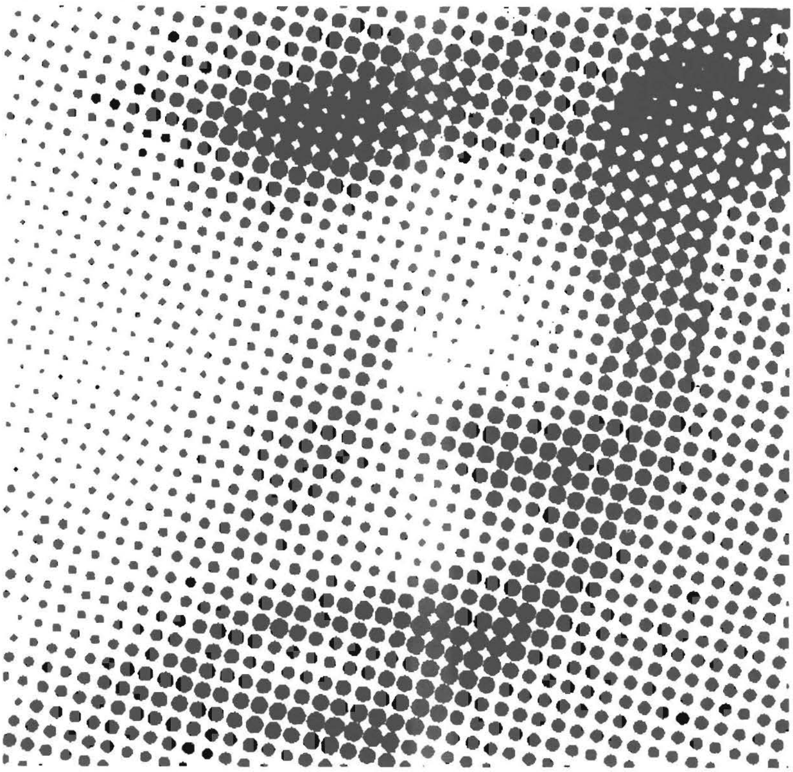

Figure 12. Reconstruction in four blocks using polynomial grid. Upper right block is the original bit map.

Comparing the method with the MMR code and Usubuchis method the data compression is higher, but the code is not information loss-less. However it provides a simple implicit edge detection, preserving the edges.

An information loss-less code could be obtained with a 'second-channel' coding the errors (Chao, 1982). The second channel could be applied in cases where results of the first channel is not satisfactory In four color reproduction, especially using grey component replacement, the second channel need only be used for the black separation, which gives most of the contrast.

The algorithms described only uses very simple operations at pixel level. The only operations at pixel level are selecting screen cells or screen cell triangles, which may be stored in a look-up table, and counting the number of black or white pixels. At screen cell level a few multiplications are required, but the operations are still simple. Therefore the described methods, some of

Figure 13. Reconstruction in four blocks using polynomial grid. The original was decompressed following the grid.

which are implemented in software, are suited for implementation with fast dedicated hardware.

7. Conclusion

Compression rates above 20 are achieved in halftone pictures. The overall compression rate for the four color separations may be as high as 80. Higher rates may be achieved by employing normal image coding techniques to the coded grey-values. The method has an implicit edge detection, preserving edges. The new method for describing the screen has the speed of look-up table screening, but may operate at arbitrary screen angles and ruling. Special treatment of high contrast areas would allow for rendition of high contrast. The method following the grid structure when descreening avoids noticably degradetion of the image, while the algorithms are simple and suited for implementation with fast dedicated hardware.

Further work include trials at a larger scale and rescreening with dot shapes similar to the original.

Acknowledgements

We thank the people of Eskofot A/S, who delivered the scanned halftone picture. The picture was scanned with an early version of the Eskoscan 2540.

References

Allebach, J.P.
1979. "Aliasing and quantization in the efficient display of images", J. Opt. Soc. Am., Vol. 69, No. 6, June 1979, pp. 869-877.

Bresenham, J.E.
1965. "Algorithm for computer control of a digital plotter", IBM Systems Journal 4, No.1, pp. 25-30.

Baird, H.S., and Forchhammer, S.
1988. "The Alignment Angle of Point Sets", in (Forchhammer, 1988)

Bodson, D, Urban, S.J., Deutermann, A.R., and Clarke, C.E.
1985. "Measurement of data compression in advanced group 4 facsimile systems", Proc. IEEE, pp. 731-739.

CCITT
1984."Recommendation T.6, Facsimile Coding Schemes and Coding Control Function for Group 4 Facsimile Apparatus".

Chao, Y.
1982. "An Investigation into the Coding of Halftone Pictures", (M.I.T.) Ph.D. Thesis.

Forchhammer, M.
1987a. "Data Compression of Documents with Text and Halftone Pictures", (Technical University of Denmark), Master Thesis, (in Danish).

Forchhammer, S.
1987b "Skew and distance of digitized grid points in the plane", Proceedings, 5th Scandinavian Conf. on Image Analysis, Stockholm, Sweden, pp. 451-458.

Forchhammer, S.
1988. "Representation and Data Compression of Two-Dimensional Graphical Data", (Technical University of Denmark), Ph.D. Thesis (in spe).

George, H.F., and Toor, Y.
1983. "Halftone scanning for electronic gravure cylinder engraving", Advances in Printing Science and Technology, Vol. 17. pp. 170-185.

Johnsen, O., Segen, J., and Cash, G.L.
1983. "Coding of two-level pictures by pattern matching and substitution",

BSTJ, Vol. 62, No. 8, pp. 2513-2545.
Jordan, B.J.
1985. "An update on Crossfield data compression technology",
Taga Proc., pp. 96-107.
Kekolahti, P
1982. "The relationship between digital screening methods, number of pixels per picture and newspaper picture quality"
Lasers in Graphics: Publishing in the 80s, Conf. Proc., Florida, pp. 275-284.
Kekolahti, P., Laihanen, P., Nasanen, R., and Saarelma, H.
1983. "Digital screening in newspaper reproduction",
4 (Otnäs), (In Swedish).
Niblack, W.
1985. "An Introduction to Digital Image Processing",
(Strandberg, Birkerød).
Schreiber, W.F.
1986. "Fundamentals of Electronic Imaging Systems",
(Springer-Verlag, Berlin Heidelberg).
Stucki, P.
1979. "Image processing for document reproduction",
Advances in Digital Image Processing, (Plenum Press New York), pp. 177-218.
Usubuchi, T. Omachi, T., and Iinuma, K.
1980. "Adaptive predictive coding for newspaper fascimile",
Proc. IEEE, pp. 807-812.
Wilson, M.R.
1986. "Machine vision in the graphic arts: Digital registration of four color halftone separations",
Taga Proc., pp. 525-542.