DTU Library

Algorithms for coding scanned halftone pictures

Forchhammer, Søren; Forchhammer, Morten

Published in:
9th International Conference on Pattern Recognition

Link to article, DOI:
10.1109/ICPR.1988.28226

Publication date:
1988

Document Version
Publisher's PDF, also known as Version of record

ALGORITHMS FOR CODING SCANNED HALFTONE PICTURES

Søren Forchhammer and Morten Forchhammer

Institute of Telecommunication, 343,
Technical University of Denmark, DK 2800 Lyngby, Denmark

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 with 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. The pages are scanned at high resolution creating a binary image. Presently, often the CCITT Facsimile Group IV MMR (modified Modified R E A D) code ([1]) is used for data compression. Other coding schemes have been devised to achieve exact coding of scanned halftone pictures [2], [6], [8].

We propose a non exact coding, which achieves higher compression rates. A page is divided in blocks of say 256 by 256 pixels. Each block is segmented to be a text/graphics or a halftone block. The text/graphics and halftone blocks are coded 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 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 descreening method proposed and some variants are given in Sections 2 and 3, with emphasis on the handling of the screen. Approximate compression rates for these methods are given in Section 4. In Section 5 results of one implementation within the general coding format are presented.

2. Describing the Halftone Screens

2.1. 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:

$(r(x,y);c(x,y))$, $(x,y) \in R^2$ 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 (Figure 1). 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 $(V_{1x},V_{1y},V_{2x},V_{2y}) \in R^4$

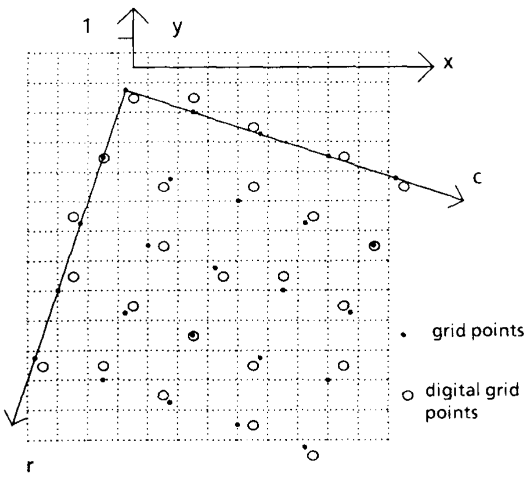


Figure 1. Digital grid points.

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 at the corners (with subscripts a, b, c, and d at (x,y)-coordinates (0,0), (N,0), (0,N), and (N,N) respectively) controlling the grid, the (r,c) coordinate system may be described by a polynomial description. This gives an indirect description of the centerpoints' (x,y) coordinates:

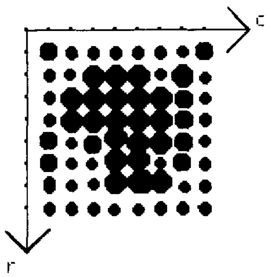
$$\binom{r}{c} = \binom{r}{c}_a + \left[\binom{r}{c}_b - \binom{r}{c}_a\right]\frac{x}{N} + \left[\binom{r}{c}_c - \binom{r}{c}_a\right]\frac{y}{N} + \left[\binom{r}{c}_d - \binom{r}{c}_c - \binom{r}{c}_b + \binom{r}{c}_a\right]\frac{xy}{N^2} \tag{3}$$

CH2614-6/88/0000/0297$01.00 © 1988 IEEE

2.2. 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. 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.



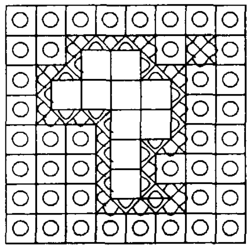


Figure 2. Bit map

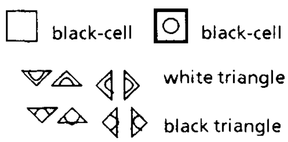


Figure 3. Corresponding segmentation in cells and triangles.

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 of black and white partial screen cell triangles (see Figures 2 and 3). Given the grey values in the digital black (or white) screen cells it is possible to segment the picture into shadow and highlight areas.

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 [3], [4].

2.3. Estimating Screen Parameters

It is necessary to estimate the screen parameters from the halftone image when using the data structure above. A near accurate estimate of the screen parameters may be known beforehand, given by an operator, or obtained from a Fourier based estimation technique [3].

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).

3. 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 [5]. The method implies a low pass filtering.

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 of descreening 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 [7]. This gives a size independent description which the user may use to regenerate the dots, given the type of shape they have.

4. Data Compression

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

$$R = \frac{\lceil \log_2 n \rceil}{n} \qquad (4)$$

where n is the number of pixels per screen cell and R is the compression ratio (the inverse figure of the compression rate).

The result of the descreening of Section 2.2. 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 Section 2.2. The grey value of each cell/triangle has a specific interval within which the value is coded. A formal description is given in [4]. 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.

5. Results

The four 256·256 blocks of the 512·512 test picture of Figure 5 were coded as described above. The same ordering of a-d as for the corners of the blocks is used. The results are given in Figure 4.

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 4. Compression rates. Column 2 and 3 gives result of a code based on Section 2.2. Column 4 gives the result of coding the grey values.

The halftone in Figure 5 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. Compared with this 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 rate, when storing border values.

Figure 5 Binary image of the original 512 by 512 scanned halftone picture.

Figure 6. Reconstruction in four blocks using polynomial grid. The original was descreened following the grid.

The compression rate may be increased by employing conventional image coding, as predictive or transform coding, to the grey values of the image.

In Figure 5 the original binary picture is shown after scanning and thresholding but before compression. Treating each halftone block separately using a linear grid may give visually disturbing results, if the original grid is varying, when putting neighboring blocks together. Using the polynomial grid description of Section 2.1 secures 'phase continuity' across the block borders giving a satisfactory visual result (Figure 6). Also when putting a block of the original bit map together with blocks re-screened with a polynomial grid, the result is satisfactory [3]. This might occur if a block with (part) halftone is coded using MMR.

In Figure 6 the descreening was done following an estimate of the grid in the original. Figure 6 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 Section 4. On the other hand the distortion of the shape of the dots would be reduced in this area with the method of Section 4.

The algorithms described only use 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 which are implemented in software, are suited for implementation with fast dedicated hardware.

6. Conclusion

Compression rates above 20 are achieved in halftone pictures. 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. 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.

References

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

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

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

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

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

[6] B.J. Jordan, "An update on Crossfield data compression technology," *Taga Proc.*, pp. 96-107, 1985.

[7] P. Stucki, "Image processing for document reproduction", in (ed. P. Stucki), Advances in Digital Image Processing, New York, Plenum Press, pp. 177-218, 1979.

[8] T. Usubuchi, T. Omachi, and K Iinuma, "Adaptive predictive coding for newspaper fascimile," *Proc. IEEE*, pp. 807-812, 1980.