ELECTRONIC SCREENING AT ARBITRARY ANGLES AND RULINGS

Søren Forchhammer and Kim Skovgård Jensen*

Abstract: A new method for electronic screening at arbitrary angles and rulings is presented. It is based on a new digital representation of halftone images. A mathematical formulation related to digital geometry is given, encompassing some variatiohs and generalizations. In extension to this, analysis of digital methods for processing of halftones is given. Incorporating number theory, the analysis focuses on digital implementations and moiré and shadowing problems in this connection.

The method has been implemented on a transputer-system. Experiments on artificial and real test images have been carried out giving promising results. A sample of these is shown.

The method performs as a look-up table screening. It uses 'sub-dots' but compared to other look-up table screening methods with the same sub-dot resolution, the dots are more regular. In comparison with threshold screening, it renders a smoother image, whereas threshold screening is better at rendering line work. Extensions to improve the ability of rendering line work are given.

1. Introduction

This paper deals with the problem of digital screening, i.e. the process of converting a digital gray value (contone) image into a bi-level image. A new method of screening at arbitrary angles and rulings have been developed and implemented based on a new description of halftone grids. The description has also been used for data compression (Forchhammer, 1988b,1991) and descreening from scanned halftone images. The description is especially suited for clustered dots organized in a grid.

In Section 2, a mathematical description of screening is given. In Section 3, the new (DGP) algorithm and the implementation are described. In Section 4, a test image is screened with the new DGP method, and for comparison, threshold screening. Methods of analyzing and reducing moiré effects are discussed in Section 5. Implementing the new screening method (and similar methods) using a look-up table, the size of the table might be a problem. The order of the size is determined in Section 6. Extension of the new method is outlined in Section 7, e.g. for improving the rendering of edges and linework.

---

* Institute of Telecommunication, Technical University of Denmark

2. Halftone Screens and Digital Geometry

In the halftoning process three entities are involved. The original gray value image (G), which we want to render, is combined with the halftoning screen (S) to create the resulting bi-level halftone image (B). Each of these entities may mathematically be described as a function of two variables.

The term *pel* is used for the bi-level (black or white) picture elements (of the output plotter). The term *pixel* is used for the gray value picture elements of the original image/photograph (from the input scanner).

The halftone screen can be described in a continuous coordinate system (s,t) with integer values at the screen dot centers. The images can be described in another continuous coordinate system (x,y) with integer values at the plotter pel centers. There is a bijective relation between the two coordinate systems. A linear grid is described by the relation

$$x(s,t) = s \cdot V_{1x} + t \cdot V_{2x} + e$$

$$y(s,t) = s \cdot V_{1y} + t \cdot V_{2y} + f$$

where $(V_{1x}, V_{1y}, V_{2x}, V_{2y}) \in \mathbf{R}^4$ and $(e,f) \in \mathbf{R}^2$ gives the offset. The angle θ of the grid vectors corresponds to the angle of the halftone screen and the length $|V|$ to the inverse of the screen ruling r (relative to the plotter resolution).

Different screening methods may be characterized by the difference in the mathematical relations (mappings) of the coordinate systems of the three screening entities.

Digitizing the grid description for a look-up table (lut) treatment of the bi-level image, it is desirable to tesselate the bit-map in a way that each pel belongs to exactly one lut-element. This way each pel is addressed exactly once.

We combine the above criterias to define a class of halftoning techniques including threshold, look-up table and the new DGP screening. The characteristics are

C.1. The structure of the halftone image is a grid
C.2. Each pel in the output is addressed (/considered) exactly once.

This class may be described by the following logical decomposition:

1) Partitioning the output pels (B) in look-up table elements.
2) Interpolation of the gray values of the input image (G) to assign a gray value to each lut-element.
3) Look-up table assignment of the color to the pels of the lut-elements on basis of the gray value determined in 2) (This is equivalent to a mapping).

Ordered dither may be described within this framework, but we shall proceed with clustered dot halftoning techniques only.

Important features of a screening process includes the ability to generate (approximately) arbitrary angles and rulings, resolution of the screening, and lack of artifacts related to control of the dot areas.

Threshold screening is a digital version of conventional screening. The gray value of the original image (interpolated) at each pel (= lut-element) of the resulting image is thresholded by the value of the threshold function (S) corresponding to the screen (look-up table decision). According to (Kekolathi, 1983) the Sci-Tex Response-300 uses threshold screening. Threshold screening may use any screen, but the method does not have complete control of the gray values making it vulnerable for some screens (with rational slopes). It is recommended to offset the conventional angles by 7.5°.

In look-up table (lut) screening a number of pels are collected in an element, which has a specific dot pattern for each gray value (and relative position). Either a whole dot or part of a dot is determined at each look-up. Look-up table screening generally uses rational grid slopes. Hell uses the rational grid slopes (1/3, 1/1, 3/1, 1/0) with the corresponding angles (18.4°, 45°, 71.6°; 90°) to approximate the values of the conventional screen (Kekolathi, 1983). This lut screening ties the coordinates of the bi-level image and the screen together using a one to one mapping. The lut-elements may coincide with the input gray values making steps 1) and 2) trivial. In comparison, threshold screening relaxes the relation of the grids by quantizing the screen coordinates, modulo the screen cell, giving a many to one mapping of bi-level coordinates onto screen coordinates.

Below a different relation between the coordinate systems is presented. It is suitable for description of the new DGP screening method .

In a halftone image, black dots are observed in highlight areas and white dots in shadow areas. The black and white dots are positioned in two displaced grids. The black screen dot centers are defined by $(x(s,t);\ y(s,t))$, $(s,t)=(i,j)\in Z^2$ and similarly $(x(s+\frac{1}{2},t+\frac{1}{2});\ y(s+\frac{1}{2},t+\frac{1}{2}))$, $(s,t)=(i,j)\in Z^2$ define the white screen dot centers. Rounding of the dot center coordinate values gives the corresponding digital grid points (Fig. 1). The prescript digital denotes a discrete representation in the scanner coordinate system.

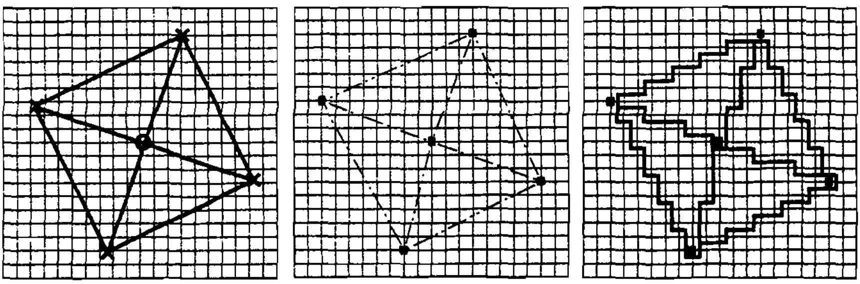

Figure 1 (from left to right) a) A cell with triangles. b) The digital grid points. c) The digital triangles lut-elements.

Drawing digital straight lines between 4-neighboring digital black grid points partitions the plane into digital white cells (Fig. 1). The digital black cells are similarly defined. These (approximating) cells do not correspond to a digitization of the lines bounding the screen cells nor do they have the uniformity in area usually expected in digital halftoning. These problems are partially circumvented by restricting the use of white cells to dominantly dark areas and black cells to light areas. Drawing digital straight lines from the digital center of a cell to the four corners will partition the cell into four digital triangles (Fig. 2). For a given position

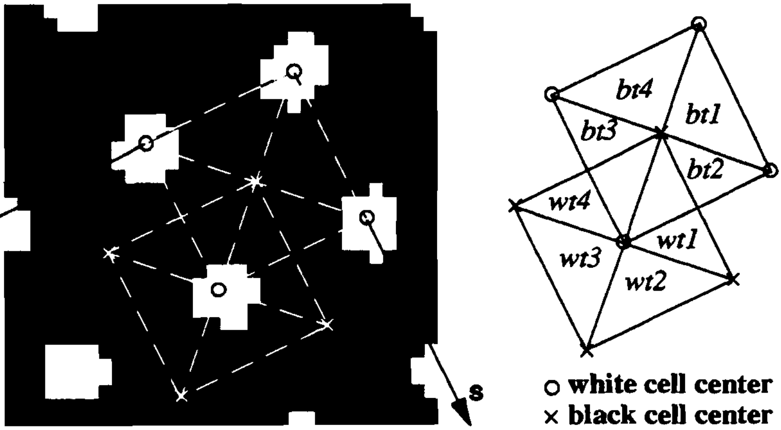

Figure 2 White and black cells and triangles.

of the digital center, the two other digital corners may take on (at most) four different digital values. This gives at most 16 different digitizations. The triangles will be used when changing the color of the cells.

The DGP screening gives a resolution of 4 (-8) sub-dots to a halftone cell.

Using a unique digitization of grid points and lines, criteria C.2 is satisfied for the black and white grids, respectively, and more generally if a partitioning satisfies C.2 in the continous domain it will also do so in the digital domain. The used combination of the black and white grids given by Figure 3 has been proven to satisfy C.2 as well (Forchhammer, 1991).

A generalization of the cell and triangle partitioning described above is to digitize the vertices of a geometric structure (e.g. a hexagon), and draw digital straight lines between the digital vertices. Any grid resolution may be used for this. This way the digital grid points (vertices) describe the new partitioning, for which reason the notion of DGP (digital grid point) is used.

The new method relaxes the relation between the grids by quantizing the (x,y) coordinates of the grid points giving a many to one mapping of screen coordinates onto pel coordinates.

An alternative is to digitize the boundaries of a partitioning, directly. This will e.g. give a true digital grid. Generation of square (sub-)cells is analyzed in (Forchhammer, 1989).

As described here within the same framework, the differences between the screening methods lies primarily in the complexity of facilitating the different features mentioned ealier in this section. It might be necessary to modify the methods deviating from the presented framework in some cases, e.g. by applying error diffusion.

Two details of the screening process have not been treated above. As mentioned, the shape of the halftone dots may be described as a two-dimensional threshold or priority function $S(x,y)$. In this paper a 2d-cosine shape description is used

$$S'(s,t) = \cos(2\pi s) + \cos(2\pi t)$$

where the coordinates of the halftone grid s and t are prescribed such that the period of S' is one grid cell. S is obtained from S' by ordering the values. This is used directly for the priority function. For the threshold function the values are prescribed relative to the range of the gray value image.

When gray values of a digital image is needed at a position different from a sample position interpolation is used. Stucki (1979) describes polynomial interpolation of order 0, 1, and 3 which are special cases of one-dimensional Lagrange interpolation using a polynomium of order N-1 given by N points.

3. Screening Algorithm and Implementation

Based on the digital grid point (DGP) description of the previous section, the new DGP screening algorithm is described below.

The triangles of the DGP grid is defined by the digital grid points at its corners. This results in triangular lut-elements of varying size and digital shape. For a specific triangle element, the gray value determines the dot size relative to the size of the triangle. Quantizing these normalized dot sizes to integer numbers of pels causes errors. To reduce the risk of moiré and false contours, error diffusion may be applied. Here error diffusion is applied among the four triangles of each cell.

Based on the gray values of the black cells, these are segmented into highlight and shadow state by thresholding at 50%. Whether to use black or white triangles is determined by the states of four neighboring cells. A state diagram is given in Fig. 3 and Fig. 4 illustrates the use of cells and triangles to cover the plane. In the remaining, a digital cell is only a logical structure and will allways be treated as four triangles.

The new DGP algorithm for screening at arbitrary angle θ and screen ruling r is given below. The operations to be performed may be divided into two. 1) The operations of setting up the look-up table (lut), given by the angle and ruling of the screen along with the halftone dot description. 2) The operations for generating the halftone image from the gray value image.

Algorithm DGP screening (G, θ, r,S)

//Given a gray value image G, the screen angle θ, the screen ruling r (relative to the plotter), and the halftone dot shape description S generates a bitmap of the screened image (B)//.

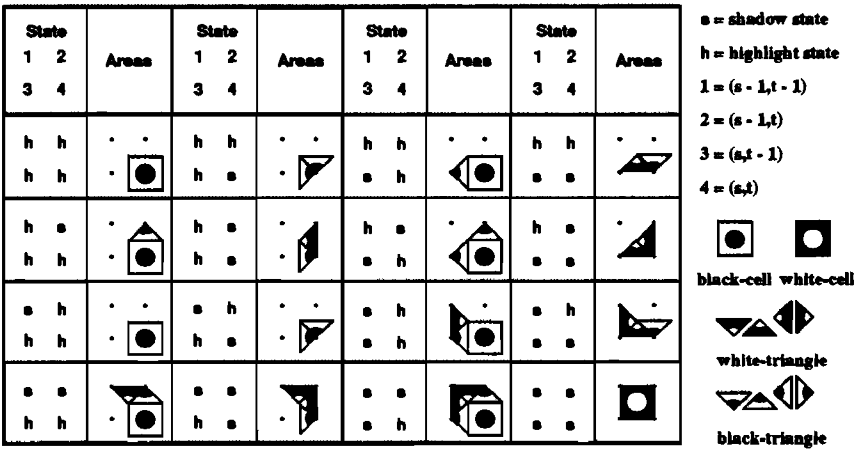

Figure 3 State diagram partitioning the image in cells and triangles.

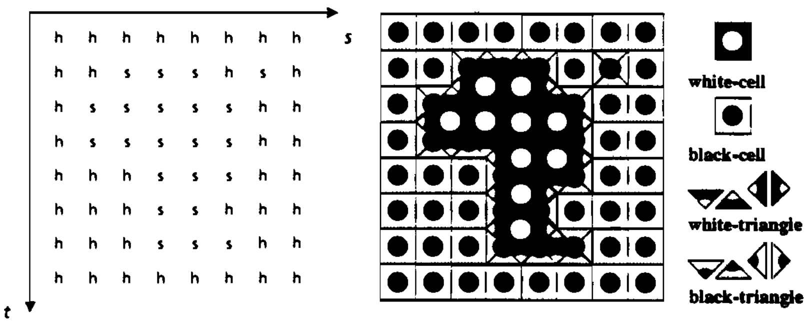

Figure 4 Left) State of black cells. Right) Corresponding coverage of plane by cells and triangles.

Set up :

S.1 Generate the digital triangular shapes of the lut-elements.

S.2 From the pel priority table based on S and the digital triangular shapes of S.1 generate the lut elements.

Operation :

O.1 Determine the gray value of each triangle by first order interpolation from the four nearest samples of G.

O.2 Calculate the cell values as the sum of its four triangles.

O.3 Segment the cell values in black and white cells at 50% dot coverage.

O.4 Determine which triangle(s) to use (see Fig. 3)

O.5 Round off grid point coordinates to determine the triangular shapes.

O.6 Normalize the relative gray value of triangle i (G_i) relative to the pel size of triangle i (T_i), i.e. $D_i = G_i \cdot T_i$ to determine the dot area (D_i) of triangle T_i.

O.7 Insert the lut-element corresponding to D_i of O.6 and the triangle shape of O.5. (Error diffusion is used within the four triangles of each cell, passing the round-off error of each triangle on to the next).

For the DGP screening algorithm to have any practical value, an implementation of the lut (S.2 and O.7), which is fairly fast in use without demanding an unrealistic amount of storage, has to be found. There are two straightforeward implementations :

1) To have one version of each triangle mask and use shift operations to situate it at the relevant position with respect to the word boundary of the actual implementation.

2) To store each triangle mask once for each possible bit position using a minimum of two words for each line of the mask in order to cope with masks that cross a word boundary.

The first solution will be slow unless the hardware performs very efficient bit shift operations. This is seldom the case for all-purpose micro-processors. The latter solution is very fast in use, but the storage demand is very large. The table size can be calculated as $n \cdot f \cdot w \cdot l \cdot b \cdot a$, where $n=4$ is the number of triangle orientations, $f=16$ is the number of digital triangle shapes[2], w is the number of positions relative to a word boundary (equals the processor word-length), l is the number of lines in each triangle mask, $b \geq 2 \cdot w/8$ is the number of bytes needed for each line and a is the maximum triangle dot area. A more general treatment of lut sizes is given in Section 6.

Using a 32 bit word-length, an average mask length of 10 and a maximum triangle dot area of 50 leads to a table size of approximatly 8 Mbytes.

A resonable compromise can be obtained using indirect addressing together with a pattern table of the following structure (Jensen, 1989). Each unbroken bit pattern is stored in every shifted version as illustrated on Figure 5. The original lut is then replaced by a table where only the addresses corresponding to the unshifted version of a mask is stored. The x times shifted version of this basic mask is obtained by adding x to each of the stored addresses prior to the table look-up. If a word boundary is crossed the part of the mask belonging to the second word is found by subtracting w from the already calculated address. White triangles can be obtained from the same address table, combining it with a bitwise inverted pattern table.

With the same parameters as before we get an address table of 128K byte and two pattern tables of 8 Kbyte each. Comparing with solution 1) we have replaced x shift operation with a single addition and one extra table look-up. This will in general lead to a substantial gain in operation speed.

To return to the question of computational complexity, a more detailed treatment of the most time consuming part of the algorithm, i.e. the masking of the triangles into the bitmap, will be given.

The operations that together forms the step O.7 in the algorithm above can be described as follows (to make the description less complicated it is assumed that the mask will not cross a word boundary) (Jensen, 1989) :

2. It is easily implemented with 16 shapes. In section 6 we will show that this number can be reduced to 9.

...

10000000000000000000000000000000
11000000000000000000000000000000
11100000000000000000000000000000
01110000000000000000000000000000

...

00000000000000000000000000001110
00000000000000000000000000000111
00000000000000000000000000000011
00000000000000000000000000000001
00000000000000000000000000000000

...

10000000000000000000000000000000
11000000000000000000000000000000
11100000000000000000000000000000
11110000000000000000000000000000
01111000000000000000000000000000

...

Figure 5 Pattern table.

For each line of each triangle mask -

1. Determine the address of the relevant pattern element by look-up in the address table:
   adr1 = address-table(triangle-number,triangle-form,dot-area,line-number)[3]
2. If it is a white triangle, get the address of the triangle outline mask:
   adr2 = address-table(triangle-number,triangle-form,max-area,line-number)
3. Determine the pattern:
   If it is a black triangle - get the pattern by look-up in the pattern-table:
   patt = black-pattern-table(adr1)
   If it is a white triangle - get the pattern by look-up in the pattern-table and combine it with the outline pattern:
   patt1 = white-pattern-table(adr1)
   patt2 = black-pattern-table(adr2)
   patt = patt1 $\wedge$ patt2 // bitwise AND operation //
4. Insert the mask in the right position in the bitmap:
   bitmap(position) = bitmap(position) $\vee$ patt // bitwise OR operation //

Example 3.1: Determine the number of triangle mask line operations (O.7) per pel:

With a screen ruling corresponding to a $|V| = 12.3$ pel grid vector and screen angle $\theta = 15°$ we can calculate the average number of lines in each mask, l (see Fig. 2):

Triangle 1 and 3: $l_1 = l_3 = \lceil \sqrt{2}/2 \cdot |V| \cdot \sin(\theta + 45°) \rceil = 8$
Triangle 2 and 4: $l_2 = l_4 = \lceil |V| \cdot \cos(\theta) \rceil = 12$
and so $l = (l_1 + l_2 + l_3 + l_4)/4 = 10$

The average number of pels per triangle line equals $|V|^2/(4 \cdot l) = 3.78$ and the inverse of this is the average number of triangle mask line operations per pel: $(4 \cdot l)/|V|^2 = 0.264$

3. If it is not the first line of the mask, then this look-up can be simplified by adding the size of one address table element to the address of the former address table element.

From this calculation and knowledge of the time consumed for each mask line operation (which depends on the hardware used), it is possible to estimate the processing time for one output (plotter) pel.

The DGP screening algorithm has been implemented on a transputer system, consisting of 17 INMOS T800 transputers each being a 17MHz micro processor with on-chip floating point unit and advanced communication facilities.

For the purpose of parallelization, the algorithm has been organized according to a division of the (plotter) bitmap into blocks (of 256 by 256 pels). The blocks can be treated independently though some redundant screening has to be done at the block borders to ensure coverage of the bit plane. The part of the gray value image needed for screening of a block can be located from knowledge of the halftone screen parameters, the number of gray level pixels used for each halftone cell and the block position in the bitmap, and distributed to the relevant processor. In addition, it is important for the choice of an efficient parallel alorithm that the calculation time spent on each block is independent of the actual image data and therefore approximately constant for all blocks. These facts make it possible to obtain an efficient parallization with a simple vectorization scheme i.e. a balanced distribution of the blocks to the processors which perform the algorithm(/calculations).

The block structure simplifies the traversing of the bitplane, following the halftone grid inside one block at a time.

4. Image Results and Quality Evaluation

The results of DGP screening applied to a test image is shown in Figure 6. Figure 7 shows the result of applying threshold screening.

The gray value test image has a resolution twice the screen ruling, which is ca. 40 lines pr. cm. The screen ruling comes from using a screen vector of 12.3 and plotting at 1270 dpi.

To evaluate the image quality, the following four qualitative metrics are used (Stoffel, 1981):

- global tone gradation
- local tone gradation (details in highlight, shadow and midtones)
- sharpness at edges
- evenness/uniformity characterized by the absence of artifacts, disturbances, and Moiré.

The screening proces is an interaction of its elements, here, the scanning of the gray value image, unsharp masking, interpolation of gray values (for DGP screening) and the plotting. Therefore, one test image gives an idea but not the complete picture of the quality of a screening method.

Threshold screening is best at rendering line work, with a remarkable sharpness of line contours, whereas DGP screening (partly due to the first order interpolation of gray values) filters the image giving a smoother image. In the test image a high degree of USM (unsharp masking) was applied. This causes the threshold screening to get an

Figure 6 DGP screened image.

almost linelike appearance at the edges, which may be considered overdone in this case.

5. Analysis of Quantization Effects on Quality

Moiré effects of screening methods are generally most visible in areas of constant luminance. (We will not here consider moiré effects dependent on the texture of the original image.)

For a screening process with rational grid vectors (with $V_1 = (p_x/q, p_y/q)$ for a regular grid) we have a two-dimensional period of the digitization process with area q. The period in each coordinate is a parallelogram described by two integer vectors in the halftone coordinate system. To the extent that, for a constant gray value, there are relative dot area variations within the period we get a moiré effect with this period. If irrational grid vectors are used, quasi-periods occur. The quasi-periods may be related to the one- or two-dimensional continued fractions of the vector coordinates

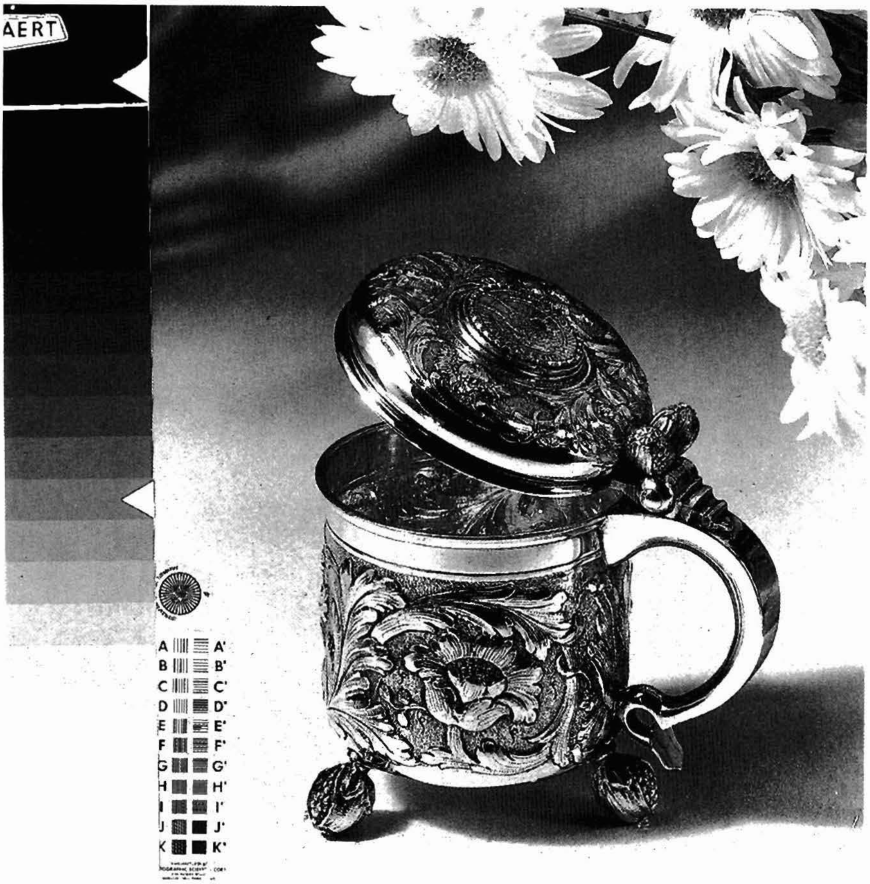

Figure 7 Threshold screened image.

(Forchhammer, 1988a). For a rational grid there will also be quasi periods corresponding to rational approximations with smaller denominator. The better the approximations, the longer the extent of the quasi period, and thereby the higher the risk of moiré to occur. The analysis of the periodicity has to be combined with an analysis of the pattern of variations within the period to give a full analysis of this moiré effect.

In case of large periods, it will be the possible moiré within the period that might show.

Example 5.1: At 45° one of the vectors in screen coordinates is (1,1), i.e. the diagonal of a grid cell, regardless of the screen ruling. This way we get a one-dimensional structure of rounding up and down the grid point coordinates.

a) With a screeen ruling corresponding to a $|V_1| = 12.7$ pel grid vector, the diagonal of a grid cell becomes $12.7 \cdot \sqrt{2} \sim 17.9605 \sim 17\ 24/25$ which gives a 449 pel quasi period (see Fig. 8.a).

b) For $|V_1| = 12.5$ the figure becomes $12.5 \cdot \sqrt{2} \sim 17.678 \sim 17\ 2/3$ giving a 53 pel quasi period (see Fig. 8.b).

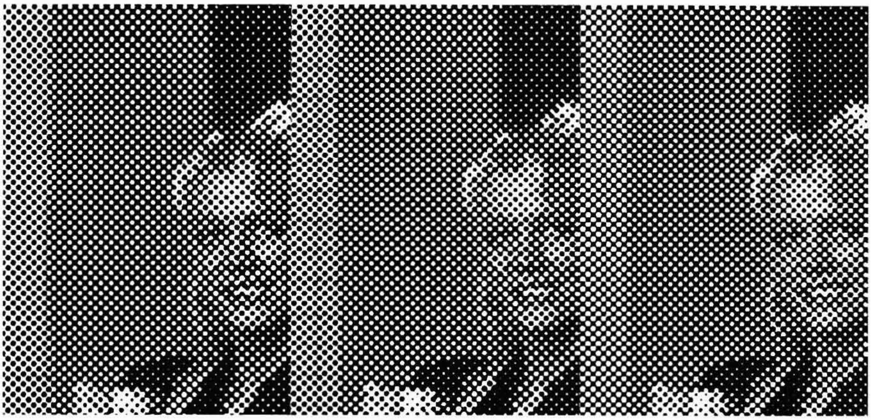

Figure 8 Shadowing and moiré effects in halftones. Screening at 45° with a ruling (left to right) a) 1/12.7 pels and b) 1/12.5 pels. Periodic effects may be noticed. The effects of b) are eliminated in c) using normalization.

Example 5.2: Screening with a screen angle of $\theta = 15°$ and a ruling corresponding to $|V_1| = |V_2| = 12.5$ pels we have the grid basis vectors measured in pels $(12.5 \cdot \cos 15°, 12.5 \cdot \sin 15°)$ and $(-12.5 \cdot \sin 15°, 12.5 \cdot \cos 15°)$, where $12.5 \cdot \cos 15° = 12.0741$ and $12.5 \cdot \sin 15° = 3.2352$. For $(s,t) = (3,1)$, $x(s,t) = 32.9870 \sim 33$ which giving a quasi period of $(s,t) = (3,1)$ for the x-coordinate and the orthogonal $(s,t) = (-1,3)$ for the y-coordinate (see Fig. 9).

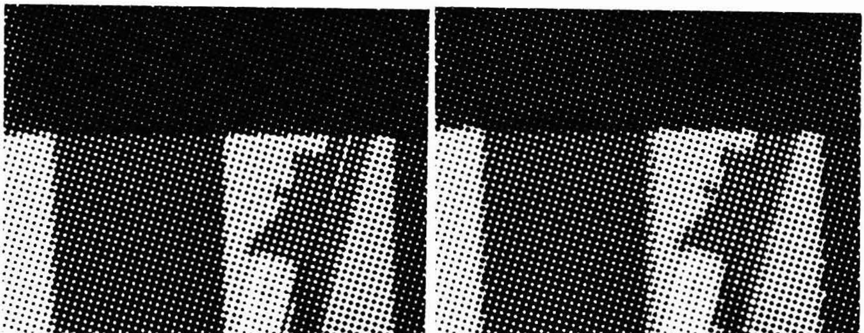

Figure 9 Left) Threshold screening: A moiré pattern may be noticed at the top of the image. Right) DGP screening of the same image using the same screen

The moiré of example 5.1 was eliminated by normalizing the triangle value and applying error diffusion within the cell (Fig. 8.c). Other deterministic methods are to pre- or post-proces the look-up table elements by e.g. deviating from a fixed screen function.

Some measure of moiré reduction would be desirable in the threshold screened image in Figure 9. The artifact does not occur in the DGP screened image.

More stochastic approaches include a randomization of (addition of noise to) the gray values and/or the grid coordinates.

The deterministic methods are to be prefered as they are to a certain extend 'predictable'. If they are not sufficient, the stochastic methods may solve the problem.

6. Complexity

The complexity of an algorithm may be measured in terms of storage requirements and computational complexity. As illustrated in Section 3, it is possible to make a trade-off between these two terms.

This section will focus on the size of the look-up table and there-by the storage requirements of the proposed screening methods. The size of the look-up table is related to the digitization(/quantization) of the (geometric) structures. Therefore, a few concepts from digital geometry will be introduced before proceeding (Forchhammer, 1989). The digitization process of a continuous object(/structure) onto a set of digital points (values) is a many to one mapping, i.e. several different continous objects maps onto one set of digital points. This also accounts for the loss of information in the digitization process. A digital object is a set of digital points possibly with some attributes e.g. that the preimage is a square. All the preimages of one digital object is called the *domain* of the object. On one hand, the digitization process is a many to one mapping. On the other hand, translating and digitizing a continous object may result in digital objects different under translation. Digital sets different under translation have different *shape*. The number of shapes could correspond to the number of entrances to a look-up table.

One important geometric structure in screening is the grid. In threshold screening the pel (grid) coordinates are digitized onto a sub-halftone grid. In DGP screening, the halftone grid points are digitized onto the pel grid. Having n continous grid points there are at most n^2 different shapes. For the four corners of a square this gives 16 shapes. The number is 9 for the three corners of a triangle (the 16 triangle shapes used in our look-up table are more than enough, thereby allowing for the use of a polynomial grid with a slight variation). Dividing the halftone cell into polygons would give at most the summation of n^2 for each polygon entrances in the look-up table, e.g. dividing the cell into four triangles gives $4 \cdot 3^2 = 36$ look-up table shapes (though they may be obtained by rotation of a multiple of 90^o from the 9 basic shapes). If the digital grid points defining the look-up table elements do not align with the halftone grid, then for each shape the different positions relative to the halftone grid have to be represented.

As mentioned in Section 2, instead of using the digital grid points to describe the lut elements, digitization of the (grid) lines of a mathematical description could be used. Analysis of this digitization and fast methods for the square (sub-)cells of a regular grid is given in (Forchhammer, 1989). Let p denote the side length of the square measured relative to the digitization grid e.g. the pel grid. The number of shapes is asymptotically upper bounded by $4 \cdot p^2$, a figure that seems to hold well for smaller squares when the angle is not rational.

Example 6.1: In Figure 10, the number of shapes for a square oriented at 15° is shown as a function of p. The number is very close to $4 \cdot p^2$ e.g. considering integer side lengths in the range 4-20, the number is $4p^2+4$ for p= 5, 9, 11, 13, and 15 and exactly $4p^2$ for the rest. With a screen ruling of 12 pel and four subcells to a cell, $p=6$ for the sub-cell resulting in 144 element shapes for each of the four sub-cells. (Each of these will again take on ca. 36 + 1 different dot areas).

The function is piecewise monotonic, dropping for certain values of p. These values correspond to the situation where the square may be placed with more than one digital point, at the same time, exactly on the edge.

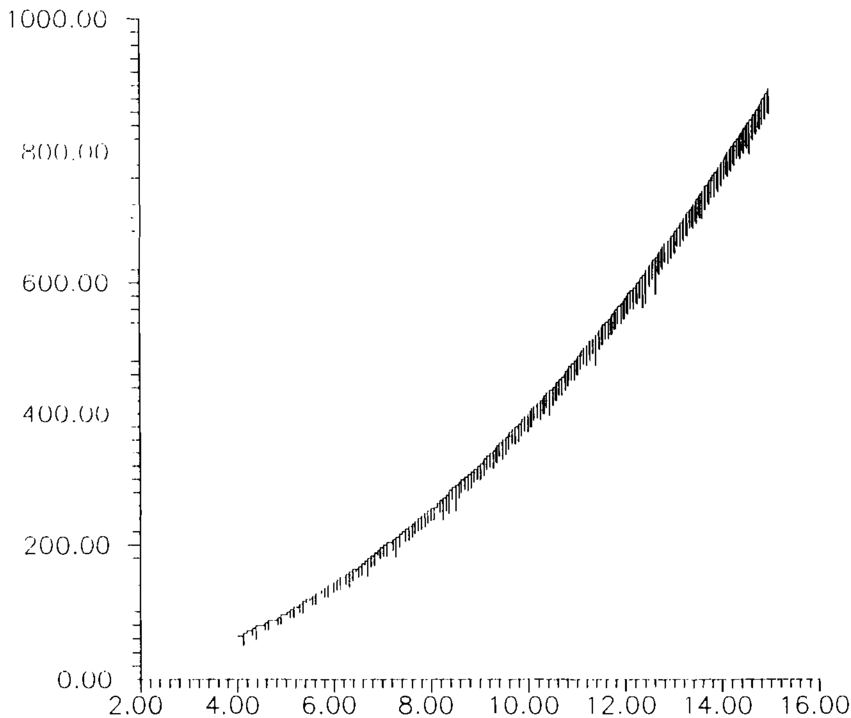

Figure 10 The number of different digital shapes for a square as a function of the side length p for a given angle of 15°.

One interesting thing about the number is that a sub-division of the halftone cell in smaller squares will not increase the total number of shapes as it is proportional to the area. Actually, the size of the total look-up table becomes smaller as each element becomes smaller. Fast algorithms to determine the number of shapes for given angle and side length are also given in (Forchhammer, 1989).

The screening process might be speeded up by using rational approximations which allows for the use of integer arithmetic. With a grid basis with rational vectors having common denominator q on irreducible form the number of shapes is strictly upper bounded by q^2 as the shapes are related to the fractional part of the coordinate values. This gives two upper bounds on the number of square shapes in a given implementation.

The DGP screening process may also be speeded up using rational approximation and integer arithmetics.

7. Extensions

A number of extensions to improve the resolution and/or the reproduction at the edges is proposed below.

If higher resolution of the screening method is needed, the halftone grid points may be generalized to any set of points giving a limited number of look-up table elements. The most tractable would probably be

1) points of a regular grid oriented as the halftone grid with a fixed (quadratic) number of points and thereby elements to a cell. A partition of the cell into 4, 9 or 16 sub-cells/quadrangles could be reasonable.
2) the same as in 1) but using triangular subcells. Here the cell could be divided into a power of 2, e.g. 2, 4, 8, 16 subcells.

Using look-up table screening with elements which are not symmetric with respect to the center, it becomes more difficult to control the gray value if the high resolution is to be maintained, with threshold screening as the extreme example.

So far the shape of the lut element and one gray value has been the input of an element to determine the pel pattern. This may be modified to imitate the effect of higher resolution. Using black and white triangles, the decision rule of using two white or two black triangles covering the same area (as in Fig. 11) may be that the line between the triangles should be perpendicular to the direction of the largest of the two derivatives (in the s and t direction) in order to follow the possible edge. This idea may also be used within the lut elements using the direction of the gradient at edges as an input, having directional elements possibly combined with a reduced number of gray values. Other possibilities would be to overwrite the triangle elements of different color or to use threshold screening at edges. This would be an image dependent screening.

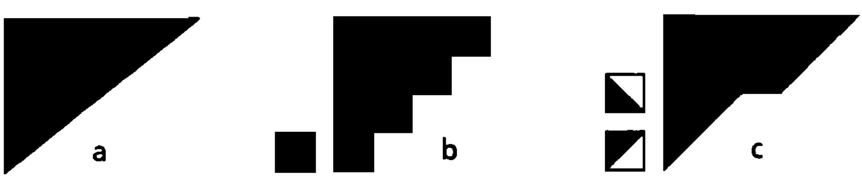

Figure 11 Ideal screening of line work (a) using square elements (b) and triangular elements (c).

When the digital grid points are used to define the look-up table elements, other grid descriptions as a (2. order) polynomial grid description (Forchhammer, 1988b) could be used and as mentioned, this will not complicate the look-up table elements much.

The polynomial grid could reduce the risk of moiré under certain conditions by introducing randomness at grid point level.

8. Conclusion

A new method of digital screening has been developed. It is based on the digitization of grid points. In some sense it is a hybrid of threshold and look-up table screening. By quantizing grid points it is possible to specify arbitrary angles and rulings as in threshold screening. The quantized grid points are used in combination with the gray values to form the argument of a look-up table, which offers better possibilities of controlling the gray values.

This combination may be advantageous in some cases. One drawback of the method is that it is more complex than the two other methods.

In shadow and highlight, the DGP method has a resolution equivalent to eight samples to a halftone dot. For edges (and midtones), the resolution is closer to four samples per halftone dot. The new method may be improved by special treatment at edges e.g. by introducing directional elements at edges.

Acknowledgement

We would like to thank A. Frøslev-Nielsen and H.O. Jørgensen of the Graphic College of Denmark for providing the test images, evaluation of the results and discussions during the project.

References

Forchhammer, S.

1988a. "Representation and Data Compression of Two-Dimensional Graphical Data", Ph. d. thesis, Technical University of Denmark.

1989. "Euclidean distances from chamfer distances for limited distances", *Proc. 6th. Scand. Conf. Image Analysis*, Finland.

Forchhammer, S. and M. Forchhammer

1988b. "Algorithms for coding scanned halftone pictures", *Proc. 9th ICPR*, IEEE, Rome, Italy, pp. 297-299.

Forchhammer, S. and K. S. Jensen

1991. "Data compression of scanned halftone images," TR-IT-127, Technical University of Denmark.

Jensen, K.S. and B. Mukherjee

1989 "Digital Systems for Data Compression of Images", (in Danish), M.S. thesis, Technical University of Denmark .

P. Kekolahti, P. Laihanen, R. Nasanen, and H. Saarelma

1983. "Digital Screening in Newspaper Reproduction", 4, Otnäs, (in Swedish).

Stucki, P.

1979. "Image processing for document reproduction," in *Advances in Digital Image Processing*, New York, Plenum Press, pp. 177-218.

Stoffel, J.C. and J.F. Moreland

1981. "A survey of electronic techniques for pictorial image reproduction", *IEEE Trans. Commun.* vol. COM-29, pp. 1898-1925.