

Redistributing Dot Connections to Avoid Visible Tile Replication in AM Halftone Screens

Rudi Bartels
Agfa Gevaert N.V.
Mortsel, Belgium

Abstract

Most halftone screens are designed to get nicely distributed dots in highlights and shadows. The place where growing dots touch each other however is often not controlled. Depending on parameters such as tile size, screen angle and ruling, the dot centers will be positioned differently on the discrete image matrix. This makes that in some regions of the tile, dots will already touch while in other regions they are still separated. Also depending on the spot size of the imaging device used, this will result in slightly lighter and darker regions in one tile, which becomes visible when the tile is replicated. This paper describes a method that uses morphological filters and operators to find the regions around the connection points. The thresholds in those regions are rearranged so dot connections become nicely distributed over the tile for all levels.

Introduction

Most reproduction devices are not capable to reproduce a continuous range of tones. They can either print ink or not. Several techniques are developed to simulate contones on this type of devices. Those simulated contones are called halftones. Halftones are obtained by printing more or less black points in the same area on paper. A distinction can be made between two major classes of methods for distributing a given number of dots over the area. FM screening, stochastic screening or dot dispersed screening tries to spread the printed dots as homogeneous as possible over the surface. AM screening or dot clustered screening will group single printed dots together in larger dots. The darker the image is, the bigger those dots will be. This paper will describe a method for enhancing the image quality for halftones obtained via AM-screening.

In the past AM-screens were generated by placing gratings between the original and the film to be lit. Currently most screens are made by using digital imaging devices. The image obtained by such a device consists of a large number of pixels that can be either on or off. Those pixels are placed in a large matrix. Every pixel is addressable by row and column number.

Many devices interact with the computer via a Postscript™ interface. In order to handle high resolutions Postscript™ is equipped with a tile based mechanism that can render a continuous tone image into a halftone matrix in a very effective way. A tile is a set of thresholds to which the continuous tone values of the image are compared.[1] If the continuous tone value is smaller than the threshold, the pixel is made black. In the other case the pixel remains white. This tile is repeated in the horizontal and the vertical direction so that the whole image is covered.

When designing a tile one should care that the next expected row of the bottom row is the top row and the next expected column for the right column is the left column. When repeating the tile over the image the seam should not be visible.

A single tile contains more than one AM-dot. In [2] it is shown that this leads to more possible screen angles. Also the bigger the size of one tile, the more gray levels can be obtained. In order to better understand the problem we want to discuss in this paper, a description of a simple tile generation algorithm will be given.

The exact position of an AM-dot center can be any floating point number and doesn't have to coincide with the center of a matrix element. For convenience a tile is selected with, for example, 64 times 64 pixels. In theory, with such a tile, 4097 distinct continuous tone values can be reproduced. For a white continuous tone all matrix pixels will be white. For the next continuous tone one AM-dot has one matrix pixel black. For the third level two AM-dots will have one pixel black. This will go on till every AM-dot has one matrix pixel black. The first pixel for an AM-dot to be black is the matrix pixels whose center is the closest to floating point center of the AM-dot. For the next level one AM-dot will have two dots on, then two AM-dots will have two dots black, etc. For a black continuous tone all matrix pixels will be black. It is desirable to have black halftone dots in the highlights that have the same shape as the white dots in the shadow tones. A way to guarantee this is by growing shadow holes simultaneously with the black highlight dots.

A spot function is used to control the order in which the pixels are added to the halftone dots as they grow. Out of the neighbor pixels of the already added pixels this one is selected for which the spot function is the highest or the lowest, depending on the definition of the spot function. If we look at the spot function as a distance function, the pixel with the lowest value or closest pixel is chosen. The origin of the spot function is the floating point center of the AM-dot. Depending on how the floating point center is positioned in regard to the integer center of the matrix pixel, the dot shape can defer from dot to dot.

If white holes in the shadows have almost the same shape as the black dots in the highlights we expect to get a nice checkerboard pattern at 50%. When the screen angle is 15° however a perfect checkerboard pattern can never be achieved. Because the orientation of the matrix pixels differs from the screen orientation the shape of the halftone dots will always be a little bit erratic. Figure 1 shows an first example of checkerboard that is oriented at 45°, and a second one which is oriented at 15°.

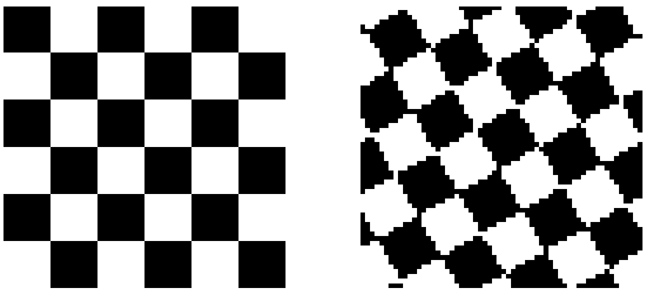

Figure 1. A screen at 45° delivers a nice checkerboard (left) while a 15° screen shows the matrix pixels (right).

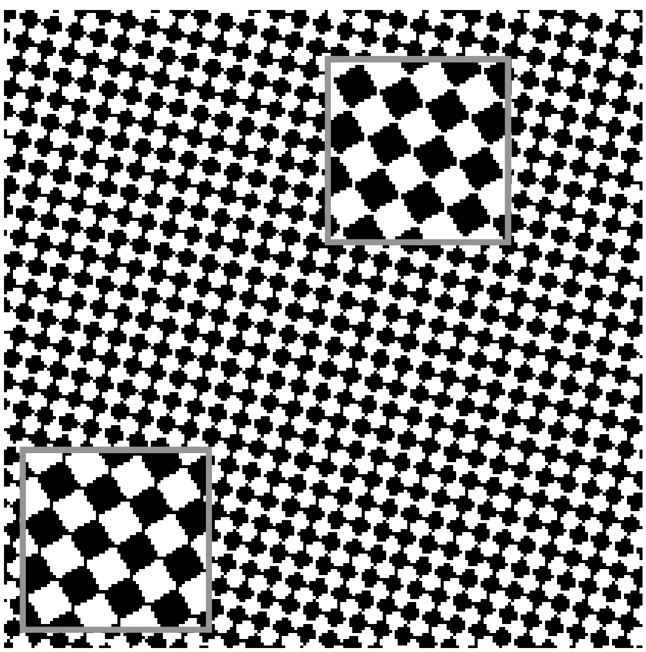

Figure 2. Example tile of 418x418 pixels. For certain combinations of tile size, screen ruling and angle, the shape of the 50% halftone dot may be different in different parts of the tile. As one can see in the enlarged parts, this can cause regions with already connected AM-dots while other regions still have isolated dots.

For certain combinations of tile size, screen ruling and angle, the shape of the 50% halftone dot may be different in different parts of the tile. Figure 2 shows two enlarged parts of a tile defining a screen of 150 lines per inch at 15° when used on an engine of 2400 dpi. As one can see, the dots are already connected in the bottom left region while they are still isolated in the top center region. As is well known in the graphic arts, dot gain may suddenly jump at tone value when halftone dots connect. This makes the bottom left region darker than the top center region. When the tile is repeated over an image the tile replication will be visible on a large 50% background. In the next paragraph an algorithm will be described to redistribute the dot connections over the tile.

Saddle Points

A dot connection is a point that touches at two opposite sides a shadow dot and at the two other sides a highlight dot. When the tile is considered as a sampled mathematical surface, a connection point is what is called a saddle point (see Figure 3). A morphologic filter can be used to find all the saddle points in the tile.

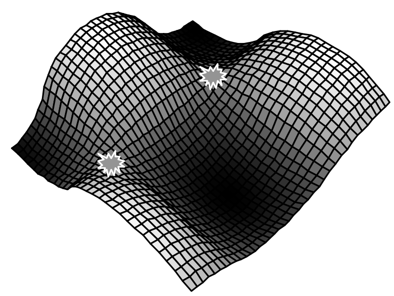

Figure 3. If tile is considered as an sampled mathematical surface a connection point is wath is called a saddle point.

The morphologic filter will look at all 8-connected neighbors of a given matrix pixel. The number of local minima and local maxima will be counted when those neighbor pixels are visited in clockwise order. The condition to have a saddle point is:

1. the number of local maxima should be exactly 2
2. the number of local minima should be exactly 2
3. the threshold of the matrix pixel should be less than the two local maxima but higher than the two local minima

Figure 4 shows the result of the above described morphologic filter on the tile of figure 2.

Histogram of Connection Points

Once the connection points have been identified, they are given new threshold values such that the connections are evenly distributed. In a first step, the number of series of connection points needs to be calculated. In a tile obtained with a euclidian spot function the highlight dots all connect around 50%. The tile of figure 5b is generated with an elliptical spot function. The dots start to touch around the 30% in one direction and around the 70% in the other direction.

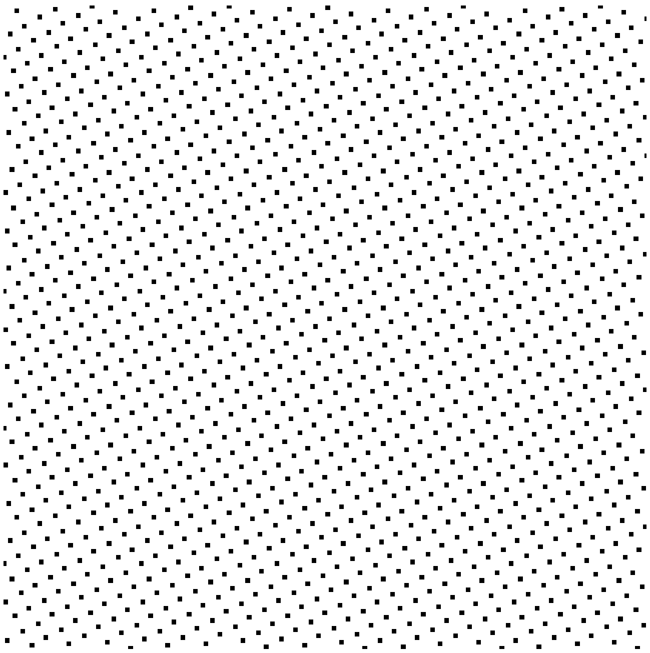

Figure 4. Result of saddle point extraction by use of morphologic filter.

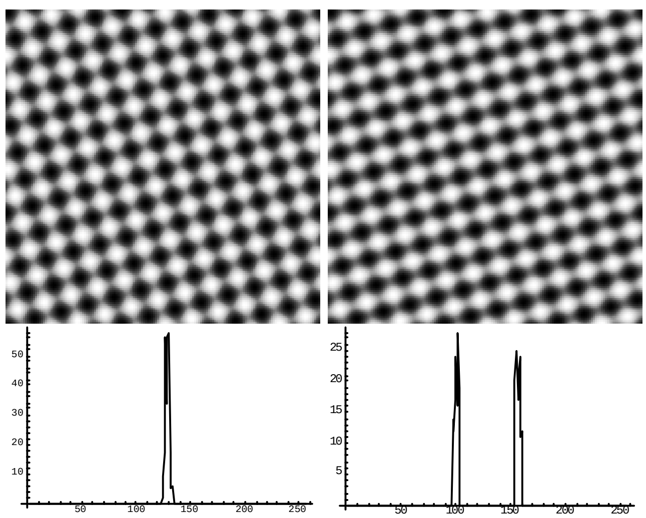

Figure 5.

In order to find the series of connection points a histogram of a downscaled threshold matrix is made. As can be seen in figure 5 the histogram for the circular spot function has only one peak while the histogram for the elliptical spot function clearly has two isolated peaks. For the latter case, the algorithm that is next described is applied for both series of connection points.

Exchange Threshold Values Between Pixels at Border and Saddle Points

The reason why in the tile of figure 2 in one region shadow dots are not yet touching is that the rhomboidal dot shape in that region is more convex, while the dots in the other region are more concave. So, if we want to alter the moment of touching, we should be able to make a concave dot more convex or a convex dot more concave. To be able to do this we have to find the dots on the border where highlight holes and shadow dots meet in the neighborhood of every saddle point. This can be done by means of another morphologic operator called dilation, with the condition that a newly added point should be at the border between highlight holes and shadow dots. This dilation will be done in an ordered way. All saddle point regions will be grown simultaneously. The first border pixel to be added will be this dot whose threshold is the closest to the threshold of the saddle point. To decide if a neighbor point is a border point we check the 4-connected neighbors of the point and see if the threshold of at least one neighbor is higher than the threshold of the saddle point and also the threshold of at least one is lower. A pixel that is added to one region can't be added to another. First one point is added to the region of every saddle point, then a second point is added to every region and so on, till no border points are left anymore.

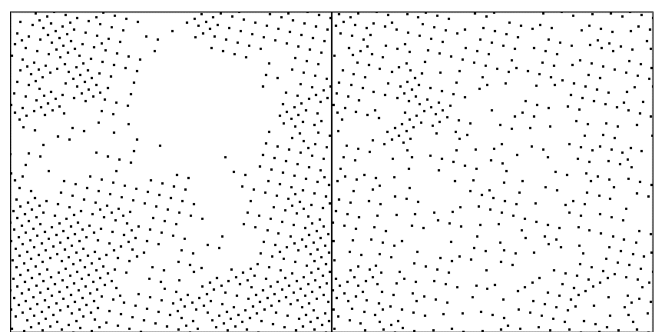

Figure 6. The left side (a) shows the original distribution of saddle point for continuous tone level 128, the right side (b) shows the result after the redistribution.

Once all the saddle point regions have been identified, we can start altering the moments of connecting. First we have to decide in what order connections should occur. We can use a Bayer-like[3] method to find optimal distribution of the connection points over the tile. To get the new threshold value for a saddle point, we use a linear interpolation between the lowest saddle point threshold and the highest saddle point threshold. For every saddle point the pixel with the nearest threshold to this new threshold will be searched. This pixel will swap its threshold with the original saddle point. Also neighbor pixels with thresholds in between the old threshold and the new threshold will swap threshold values as well. Figure 6 shows the result of the algorithm for different continuous tone values. The left image shows the original distribution of saddle points for continuous tone 50%, the right side shows the distribution after the algorithm has been applied. When output to film the visible pattern due to the unbalance of connection points has disappeared.

Conclusion

In this paper an algorithm for redistributing the connection points has been proposed. As is shown in the example, a much better distribution of connections is obtained after applying the algorithm. The algorithm can be applied to any AM-screen tile and doesn't depend on the way the original tile is generated. Further improvements can be made by controlling the shape of the newly obtained dots. Because thresholds are swapped, spikes may occur at the border of AM-dots. Another application of the algorithm that can be investigated is spreading out the moment of connecting over several continuous tones to prevent all the dot gain jumps due to the connection occurring at once. On some 16 bit halftone vignettes a line or intensity jump can be observed around the 50%. Our algorithm could make this jump less visible.

References

1. Adobe Systems Inc., "PostScript Language Reference, third edition", Addison Wessley (1999).
2. Paul Delabastita, "Screening techniques, Moiré in Four Color Printing", *TAGA, 1992 Proceeding's*, Vol.1.
3. B. E. Bayer, "An optimum method for two-level rendition of continuous-tone pictures," *Proc. IEEE Int. Conf. Commum.*, pp. 2611-2615. (1973).

Biography

Rudi Bartels got his masters degree in electronic engineering in 1987 at the Catholic University of Leuven. Till 1996 he did research in the image processing group ESAT-PSI at the same university where he designed a system for chromosome karyotyping with an artificial intelligent interface. He defended his Phd in september 1999. In 1996 he started working for Agfa Gevaert where he first implemented the ColorTune color management application on Macintosh platform. Since October 1998 he is involved in halftone screening research in the graphical division at Agfa.