Digital Multitoning Evaluation with a Human Visual Model

Qing Yu and Kevin J. Parker
Department of Electrical Engineering
University of Rochester, Rochester, NY 14627

Kevin Spaulding and Rodney Miller
Imaging Science Technology Lab
Eastman Kodak Company, Rochester, NY 14650-1816

Abstract

Multilevel halftoning (multitoning) is an extension of bitonal halftoning, in which the appearance of intermediate tones is created by the spatial modulation of more than two tones, i.e., black, white, and one or more shades of gray. In this paper, a conventional multitoning approach and a specific approach, both using stochastic screen dithering, are investigated. Typically, a human visual model is employed to measure the perceived halftone error for both algorithms. We compare the performance of each algorithm at gray levels near the intermediate printer output levels. Based on this study, an over-modulation algorithm is proposed. This algorithm requires little additional computation and the halftone output is mean-preserving with respect to the input. We will show that, with this simple over-modulation scheme, we will be able to manipulate the dot patterns around the intermediate output levels to achieve desired halftone patterns.

Keywords: multitoning, stochastic screen, human visual model, over-modulation

1 Introduction

Recently we have seen a newer and expanding role of stochastic screen in digital printing [1, 2, 3, 4] because of its implementation simplicity and visually pleasing output. The implementation of screening employs a simple pointwise comparsion, as shown in Figure 1. An input image value is thresholded by a corresponding screen value to turn the output pixel on or off. Halftone patterns from stochastic screen contain "blue noise" [5] characteristics, similar to those of error diffusion method. Since the human visual system is less sensitive to high-frequency content (blue noise), halftone patterns generated from stochastic screening are less visible to a human observer.

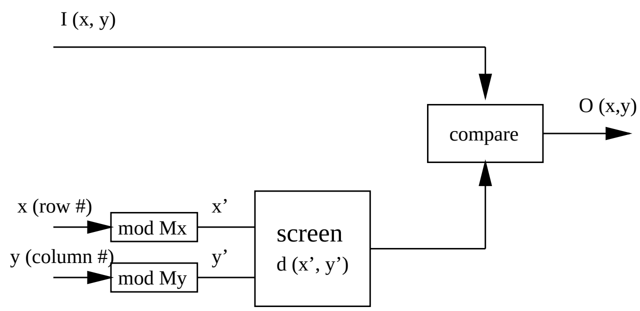

Figure 1: *Bilevel halftoning with stochastic screen.*

When devices have multilevel outputs, such as multilevel inkjet printer, stochastic screen technique can easily be generalized to utilize this new capability [3], as shown in Figure 2. It can be seen that this is equivalent to the binary implementation in Figure 1, except that the screen is first scaled to certain intermediate range before the comparison is taken, and the output is set to one of those output levels based on the comparison result and corresponding intermediate range. For example, assume the device has 4 output levels, 0, 85, 170 and 255, respectively; also assume the original stochastic screen has 256 levels from 0 to 255. If the input pixel value is 128, the screen is first scaled to the 85 and 170 range, then, based on the scaled screen value at that specific location, we either output 170 or 85. We call this simple extension the conventional multitoning scheme.

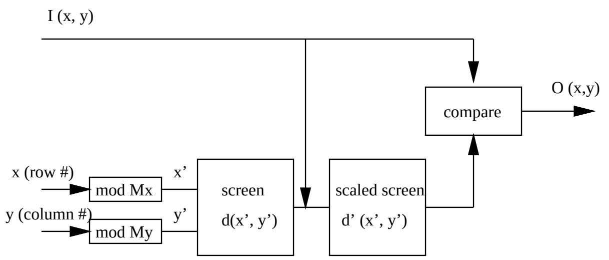

Figure 2: *Conventional multilevel halftoning with stochastic screen.*

Figure 3 illustrates the bilevel halftoned verison of a gray ramp and the multitoned version of the same ramp.

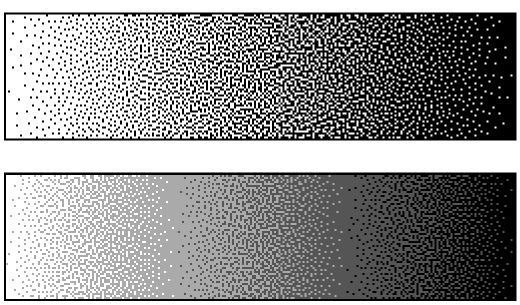

Figure 3: *Gray ramp: halftoned (above) and 4-level multitoned (below).*

2 Evaluation with Human Visual Model

To evaluate the multitoning result, a human visual model could be ultilized to study the perceived mean square error (MSE) between the original image and its multitoned version. The flow chart is showed in Figure 4. Define the input image as i(x, y), and the ouput image as o(x, y), then error e is given by the following equation:

$$e = \sum [i(x,y) * h(x,y) - o(x,y) * h(x,y)] \qquad (1)$$

where h(x,y) is the point spread function of a human eye and "*" stands for convolution.

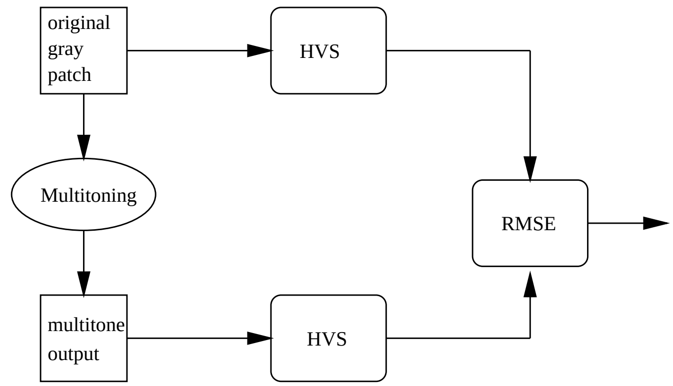

Figure 4: *Evaluation with HVS.*

In spatial frequency domain, the equation is given as:

$$E = \sum [I(X,Y) - O(X,Y)]H(X,Y) \qquad (2)$$

A model of the low-contrast photopic modulation transfer function was used to characterize the human visual system [6]:

$$H(i,j) = a(b + cf_{ij})exp(-(cf_{ij})^d), \quad if \quad f_{ij} > f_{max} \qquad (3)$$

$$H(i,j) = 1.0, \qquad otherwize \qquad (4)$$

where the constants a, b, c, d take on values as 2.2, 0.192, 0.114, and 1.1, respectively. The unit of frequency is cycles/degree.

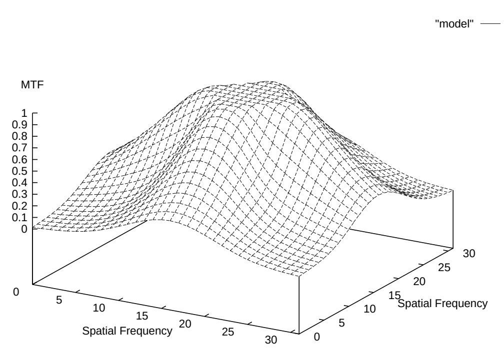

Figure 5: *HVS model.*

A plot of this visual model is shown in Figure 5, which illustrates the low-pass nature of the visual system and the reduced sensitivity at 45 degrees. To apply this model for a specific viewing distance and resolution, we need to do a conversion from cycles/degree to cycles/inch. Let P be the printer resolution, d the viewing distance from the eye to the object, N x N the support of the image, (i,j) a location in the spatial frequency domain, and f_i, f_j the spatial frequency in cycles/degree in the two dimensions. It can be shown [7] that :

$$f_i = \frac{2idP}{N} * tan(0.5°) \tag{5}$$

and

$$f_j = \frac{2jdP}{N} * tan(0.5°) \tag{6}$$

The radial frequency is given by:

$$f_{ij} = \sqrt[2]{(f_i)^2 + (f_j)^2} \tag{7}$$

To incorporate the decrease in sensitivity at angles other than horizontal and vertical, the radial frequency is scaled such that $f_{ij} -> f_{ij}$/S, where

$$s(\theta) = \frac{1-w}{2} cos(4\theta) + \frac{1+w}{2} \tag{8}$$

where w is a symmetry parameter, f_{ij} can then be substituted into equation. With this conversion, the human visual model can be applied directly on a digital image.

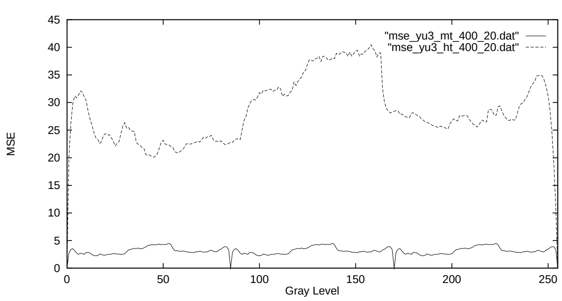

Figure 6: *MSE for bilevel and multilevel halftoning.*

Gray patches for each level are generated and multoned with the conventional scheme, and MSE is calcuated for each patch. Figure 6 shows the MSE vs gray level for both bilevel halftoning (dashed line) and 4-level multitoning (solid line). In the later case, we assume the device has 4 output states, say 0, 85, 170, and 255, respectively (0 is pure black while 255 is pure white). The stochastic screen used is 128 by 128 in dimension and has 256 levels. We set the viewing distance at 20 inches and resolution as 400 dpi (these two parameters will be used throughout this paper).

A few observations are warrented. First, there is great similarity between these two curves, each segment of the multitone curve could actually be perceived as a scaled version of the bilevel curve with compressed tone scale range. This agrees well with the implementation of multitoning as an extension of bilevel halftoning process. Second, for the multitone curve, around each intermediate output level (85 and 170 in this case), there is distinct dipping and shootup of MSE, which could also be found for the bilevel curve at both ends of the tone scale. The explaination for this is straighforward. Right at the output states, there is no halftone error introduced, which leads to zero visaul error. A little away from those levels, you end up with dot patterns having sparse minority dots over an uniform background, and human eyes tend to pick up those dots very easily, resulting in high visual error. In practice, screens are usually "punched" at both ends of the tone scale; for example, level 240 and beyond are set as white and level 15 and below as black. This works fine for bilevel halftoning, since most image detail does not fall in those levels, and punch will also increase the contrast of image. For multilevel halftoning, however, it is a different story. It could be very possible that we have an image region that contains a smooth transition just around a certain intermediate level, then with the conventional multitoning scheme, there will be a distinct texture change in the output, which will be visible within certain viewing distances, as shown in Figure 7.

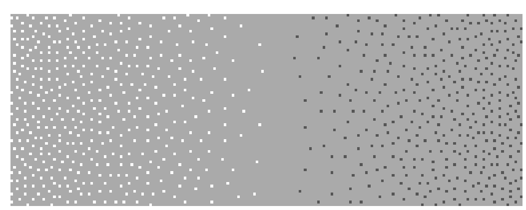

Figure 7: *Texture change around output level 170.*

Another potential problem with this approach is that some devices, such as electrophotographic printers, do not produce uniform density regions very well. For these devices, having an unifrom region in the output will actually degrade the image quality.

One thing we should keep in mind is that the conventional scheme is just a simple extension of the bilevel scheme, we have not fully taken advantage of having multiple output levels. A natural refinement of the conventional scheme would be to introduce dots of adjacent levels at gray levels near the transition level to achieve a more smooth visual transition.

3 Earlier Approach

An earlier proposed multitoning scheme [8] by Miller and Smith could be a solution for this problem. In this scheme, the modularly addressed screen is used to store pointers to a series of screen look-up tables (LUT), rather than storing actual screen values. The results of the screening process for each of the possible input levels are precalculated and stored in these LUTs. The algorithm can now be executed with only table look-ups rather than the adds and multiplies in the convenitonal scheme. Obviously, this trades off memory requirements for a faster execution.

Another advantage of this LUT-based approach is that any conceivable dot growths pattern can be specified. With the conventional scheme, as the input gray level is increased, all of the pixels in the halftoned pattern are generally increased to the second output level before increasing any of the pixels to the third output level. With this scheme, we gain the flexibility to increase the gray level of one pixel in the halftoned pattern through multilevels before starting to increase the gray level of a second pixel. Specifically, there is a texture paramter in this algorithm that controls the desgin of LUTs with a varity of characteristics from a regular screen.

We run this algorithm with the same output levels (4), same screen (128 by 128 in dimension) as specified previously, but with different texture parameters. The plot of MSE vs gray level for several texture parameter values is shown in Figure 8. A more smooth visual transition has been achieved at the expense that for large texture value, the visual error is raised over the whole tone scale. There is a trade-off here, which is not a suprise since this scheme was not optimized for our typical study. What we find is that for this study, a small texture value such as 0.0015 gives a good trade-off. For this value, what we are actually doing is that when we are far from those intermediate levels, we use the conventional scheme (bilevel) to design LUTs so that only two levels of dots are involved; when we are within a certain neighborhood of those intermediate output levels, we gradually introduce dots of adjacent levels.

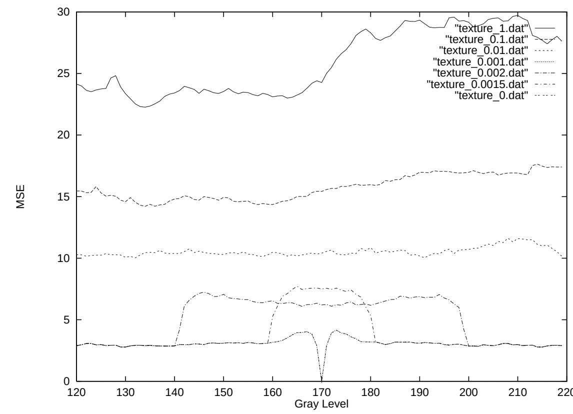

Figure 8: *MSE vs gray for different texture values.*

4 Over-modulation Approach

In this section, we introduce a novel over-modulation scheme to handle the same problem with the same goal, that is to achieve smooth transition at intermediate output levels by introducing pixels of adjacent levels. However, we would like to keep the conventional multitoning structure by introuducing a preprocessing step instead.

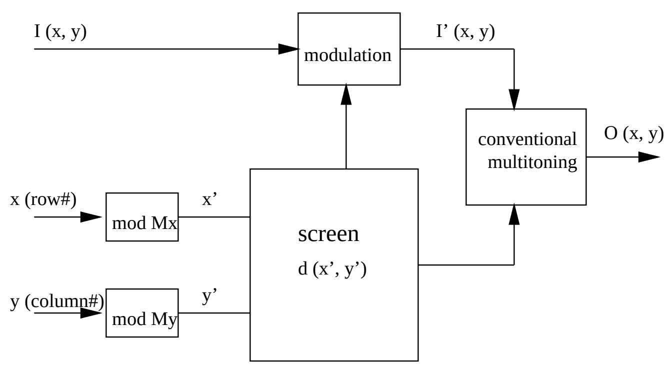

Figure 9: *Flow chart of over-modulation scheme.*

Figure 9 shows the flow chart of this algorithm. A screen-guided modulation operation is added before the conventional multitoning process. With one input pixel value, we first decide if it is inside the neighborhood of any intermediate output levels. If not, we simply output this pixel for conventional multitoning process; if the pixel is inside a neighborhood ([X-R, X+R]) of output level X, we call a modulation function, which needs the pixel value I, output level X and corresponding screen value S at that location as inputs. We modify the input pixel with the modulation to get I' and use the conventional multitoning scheme to get the output value O. The modulation process is a nonlinear one, and could be best discribed with the following pseudo computer code.

Assume I is the input value, which falls in the neighborhood of a specific intermeidate level X, and the corresponding screen value is S, then:
D = I - X
A = MAP(D)
if (D >= 0)

$$if\,(S\ >=\ 128\,)\,\{\,I' = X - A\,\}\,else\,\{\,I' = I + A\,\} \quad (9)$$

else

$$if\,(\,S\ >=\ 128\,)\,\{\,I' = I - A\,\}\,else\,\{\,I' = X + A\,\} \quad (10)$$

where MAP() is a preset mapping function (a typical one is shown in Figure 10 with range R set at 3). An example will be given in the Appendix to show that this over-modulation is a mean-preserve process.

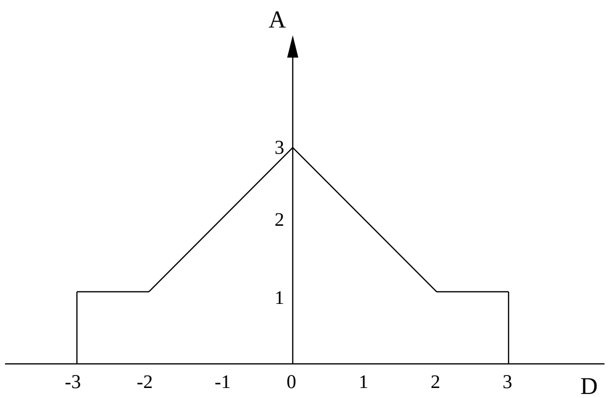

Figure 10: *Modulation function.*

In Figure 12, the second image from the top shows the over-modulation scheme on a gray ramp around output level 170 with a normal screen. For visual comparison, the conventional scheme output is also shown in the same figure at the top.

5 Over-modulation Dot Pattern Design

As we can see, around those intermediate output levels, minority dots are introduced by the over-modulation scheme to smooth the visual error transition. These dots belong to dot patterns at both ends of tone scale. However, during our regular screen design, the dot patterns at both ends are remotely correlated such that they are essentially independent from each other, therefore, it is quite obvious that a regular screen will not be optimal for our special application. If we keep this over-modulation in mind during the screen design, we should be able to add special correlation or special characteristics between these patterns to achieve better visual performance. Two methods have been proposed so far:

1. We enforce the condition that all the minority pixels (neighboring output levels) in a multilevel halftone output have a blue noise distribution such that they are maximally dispersed, as illustrated in the bottom ramp of Figure 12. This can be done in the following way. Assume we know that the top 5% and bottom 5% dot patterns will be involved in the over-modulation process. In the case of an 8-bit screen that has 256 output levels, 5% of the patterns will correspond to those of level 243 and up and those of level 13 and down. From a regular screen, we identify locations where screen values are in the range of 242 to 229, and we set these locations as forbidden ones. Starting from level 243, we construct another screen in the normal way, except that the pixels at those forbidden locations can be turned on only for dot patterns between level 0 and 13.

2. In the second method, we enforce the condition that all the minority pixels (neighboring output levels) in a multilevel halftone output will be paired, as illustrated by the second ramp from the bottom in Figure 12. We name these pairs as binars. Obviously, we could have three arrangements for these binars; they could be lined up in vertical, horizontal or diaganal directions. The screen design is very similar to the first method. With the same assumption we made for the first method, from a regular screen, we identify those locations (x,y) with screen value S(x,y) in the range of 243 to 255, then shift the locations diaganally (assume diaganal binars) and set the screen value at that locations as 255 - S(x,y). After this is done, we have built up the binar correlations for the top 5% and bottom 5% dot patterns. The rest of the dot patterns for the screen are designed in the normal way.

Based on our initial subjective tests, we find that among these three screens (regular, maximally dispersed and binars), the binars patterns perform best because they are less noisy and halftone dots are dissolved more quickly when viewing distance is in-

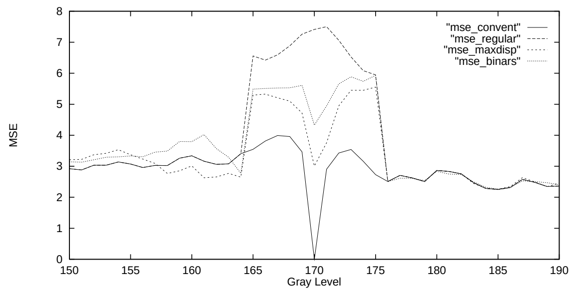

Figure 11: *MSE vs gray level for different over-modulation screens (400 dpi and 20-inch viewing distance).*

creased. Figure 11 shows the MSE curves for the multitoning outputs from these three screens. Further tests should be carried out on this subject.

6 Discussion and Future Research

There are several parameters that are very important for this over-modulations scheme, such as neighborhood range R and modulation function MAP(). Testing on different values for R and various mapping functions are worthwhile so as to find the optimal value and function.

Meanwhile, we are carrying out some psychovisual experiments to determine the "preferred" over-modulation dot patterns. Once we have this knowledge, we could put it into the screen design process. Visual experiment could also be carried out to determine, with fixed number of output levels, what the optimal levels are and in what space (code value or density) for this over-modulation scheme.

This over-modulation scheme could also be extened to multilevel color rendering, where different color dots will interact with each other around certain boundaries.

Finally, all the simulations are currently done on a dye-sublimation printer, however, the over-modulation scheme is initially intended for a wide range of printers including inkjet. Therefore, further experiments on different printing engines will be worthwhile.

7 Conclusion

In this paper, we present a novel over-modulation scheme to improve multilevel rendering around intermediate output levels. A more smooth transition of visual error has been achieved.

8 Appendix

In the following, we will show that this over-modulation is a mean-preserve process. Assume we have uniformly distributed output levels over the whole tone scale, therefore we can define these levels as 0, L, 2L, 3L, and so on up to 255. If a input pixel I falls in the range between L and 2L, and it is in the neighborhood of 2L, then with conventional sheme, the expected value for output O should be given by:

$$E\{O\} = P_{2L} * 2L + P_L * L \qquad (11)$$

Since there is no prior screen information, P_{2L} = (I - L)/L and P_L = (2L - I)/L, therefore, $E\{O\}$ = I.

With over-modulation scheme, we have D = I - 2L, where D < 0, then A = MAP(D). If the screen value S > = 128, since D < 0, then according to the over-modulation scheme, the modified input value I' = I - A. In this case, the expected value for output O will be given by:

$$E_1\{O\} = P_{2L} * 2L + P_L * L = 2I - 2A - 2L \qquad (12)$$

where $P_{2L} = \frac{(I-A)-3/2L}{L/2}$ and $P_L = \frac{2L-(I-A)}{L/2}$. If S < 128, then I' = 2L + A, therefore, the expected value for O will be given by:

$$E_2\{O\} = P_{3L} * 3L + P_{2L} * 2L = 2L + 2A \qquad (13)$$

where $P_{3L} = \frac{(2L+A)-2L}{L/2}$ and $P_{2L} = \frac{5/2L-(2L+A)}{L/2}$. Since there is a 50% chance for each screen value to go beyond 128 as well as below 128, therefore, the overall expected value for output O will be:

$$E\{O\} = 0.5 * E_1\{O\} + 0.5 * E_2\{O\} = I \qquad (14)$$

which shows that the over-modulation process is a mean-preserve process.

Note that we use only half of the intermediate range L/2 to calculate the probability in the over-modulation scheme, since we have prior knowledge of the screen value (either S >= 128 or S < 128).

References

[1] T. Mitsa and K. J. Parker, "Digital halftoning using a blue noise mask," in *ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, (Toronto, Canada), pp. 2809–2812, IEEE, May 1991.

[2] T. Mitsa and K. J. Parker, "Digital halftoning using a blue-noise mask," *J.Opt.Soc.Am. A*, vol. 9, pp. 1920–1929, Nov. 1992.

[3] K. Spaulding, R. Miller, and J. Schildkraut, "Mehtods for generating blue-noise dither matrices for digital halftoning," *J. Elec. Imag.*, vol. 6, pp. 208–230, Apr. 1997.

[4] Q. Yu, K. J. Parker, and M. Yao, "On filter techniques for generating blue noise mask," in *Proceedings, IS&T's 50th Annual Conference*, (Boston, MA), IS&T, May 1997.

[5] R. Ulichney, *Digital Halftoning*. MIT Press, Cambridge, MA, 1987.

[6] J. Sullivan, L. Ray, and R. Miller, "Design of minimal visual modulation halftone patterns," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 21, pp. 33–38, Jan./Feb. 1991.

[7] Q. Lin, "Halftone image quality analysis based on a human vision model," in Allebach and Rogowitz [9], pp. 378–389.

[8] R. Miller and C. Smith, "Mean-preserving multilevel halftoning algorithm," in Allebach and Rogowitz [9], pp. 367–377.

[9] J. P. Allebach and B. E. Rogowitz, eds., *Proceedings, SPIE—The International Society for Optical Engineering: Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, (San Jose, CA), SPIE, Feb. 1993.

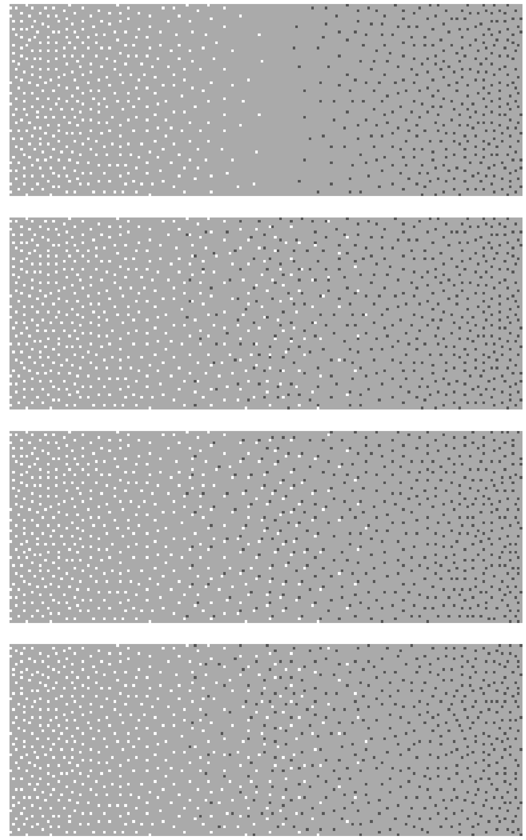

Figure 12: *Over-modulation with different dot patterns. Top: conventional scheme; Up middle: regular screen; Low middle: binars; Bottom: maximally dispersed.*