Vol. [VOL], No. [ISS]: 1–10

Versatile Decorative Halftoning

Alejo Hausner
University of New Hampshire

Abstract. This paper describes a flexible tool for decorative halftoning. Decorative halftoning produces a picture that can be seen simultaneously on two levels: either as an over-all image, or as a grid of much smaller motif images. The tool presented here is flexible: any small image can be used as a motif. When the motif is a black-and-white image, this tool improves on other techniques, which cannot handle all such images: they are restricted to motif images that can be reduced to a skeleton of curved lines. Even for such images, the tool described here retains more spatial detail from the over-all image.

1. Introduction

People have long been fascinated by pictures made up of smaller elements, such as Roman mosaics, which are composed of tiny tiles, or Archimboldo’s Renaissance paintings, which show people composed of vegetables, fruit, or books. As early as 1966, computers were used to create such pictures [1], and new algorithms continue to produce a bounty of new techniques, including photomosaics, jigsaw image mosaics and decorative halftoning. This paper focuses on decorative halftoning, and presents a simple and flexible way to create pictures which can be viewed on two levels: as an over-all image, or as a grid of tiny motif images.

Since not all readers may be familiar with halftoning, this paper begins with a brief introduction to dithering (a halftoning technique) and describes the qualities of a good dispersed-dot threshold matrix. Readers familiar with

1086-7651/06 $0.50 per page

dithering may wish to skip directly to section 2, which describes the problem this tool addresses, and a simple solution.

1.1. Halftoning

Before describing *decorative* halftoning, let’s look at halftoning itself. This term refers to a broad class of methods for reproducing full-color images on devices which are limited to only a few colors. Halftoning methods achieve this by arranging dots of a few colors into patterns that the viewer’s eye can blend into the desired color. The eye is tricked into seeing more colors than are actually there. In the simplest case, a gray-scale image must be reproduced on a device which has only two colors: black ink on white paper. To produce a printable version of this image, each pixel’s color is set to either black or white. The decision is made by comparing the pixel’s value to a threshold value. This process of restricting a large range of colors to a small range (in this case, two colors) is called *quantization.* The color of each pixel in the halftoned image is obtained by the following simple quantization rule:

Input	**Threshold**	**Output**
pixel	**value**	**halftone pixel**
$I(x,y)$	$T(x,y)$	$H(x,y) = \begin{cases} 0 \text{ (black), if } I(x,y) \leq T(x,y), \\ 255 \text{ (white), if } I(x,y) > T(x,y) \end{cases}$

There are two main types of halftoning techniques: error diffusion and ordered dither. Error diffusion takes into account the error created by quantization (*i.e.*, $I(x,y) - H(x,y)$), and “diffuses” it onto the pixel’s neighbors: a portion of the error is added to each unquantized neighbor. Variants of error diffusion differ mostly on which neighbors receive the error, and in what proportion. In the earliest form of error diffusion, Floyd and Steiberg’s [5], four unquantized neighbors receive the error. In most error diffusion methods, the threshold $H(x,y)$ is fixed at 128 (50% gray).

Dithering does not distribute error, but rather uses quantization with variable thresholds: each pixel may potentially have a different threshold $T(x,y)$ applied to it. One choice is to set each threshold to a random value, but the resulting halftone images are noisy. Ordered dither, a more popular approach, uses a fixed pattern of thresholds. For a large image, computing a different threshold for each pixel is usually very time-consuming. To save time, only a small matrix of thresholds is computed once offline, and this *threshold matrix* is then repeated across the image, just like square tiles being laid across a floor. Bayer [4] developed one of the first threshold matrices for dispersed-dot dither, which is discussed in the next section. Ulichney’s void-and-cluster matrix [8] avoids the repetitive structures in Bayer’s matrix, and is a popular choice today. Both these matrices are used in devices like inkjet printers which can produce isolated single dots of ink. When this is not possible, for example when printing on absorbent newsprint paper, clustered-dot ordered dither matrices are used. These matrices yield halftone images with larger variable-size colored dots. Unfortunately, the resulting grid of dots is often noticeable. To overcome this problem, green-noise masks have been developed [6]. They yield halftone with variable-size dots, without obvious repetition.

Halftoning continues to be actively researched, but a thorough review of existing work exceeds the requirements of this paper. Besides, such reviews already exist. For example, a recent book by Lau and Arce [7] provides a very good introduction to halftoning techniques, and also describes ways to evaluate the resulting halftone images quantitatively. In this paper, the focus will be on a single halftoning technique, Bayer's dispersed-dot dither matrices, which are simple to generate and easy to understand.

1.2. Dispersed-dot Dither

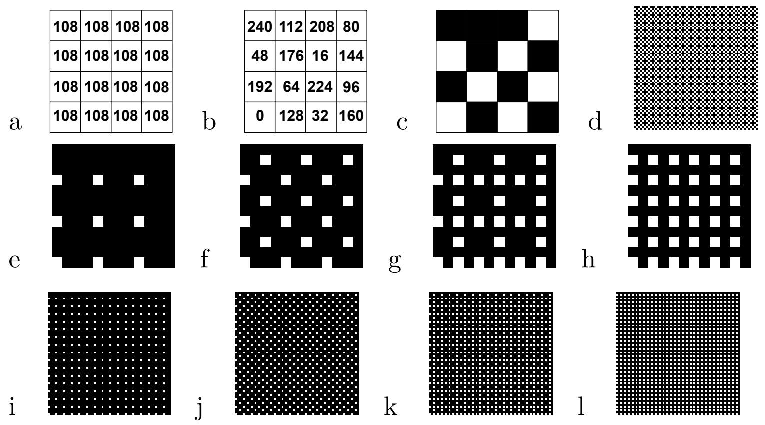
Figure 1a shows the values of a 4×4 image where every pixel $I(x, y)$ has gray value 108, and fig. 1b shows a threshold matrix T of the same dimensions, built with Bayer's algorithm. Figure 1c shows the halftone H obtained by applying the quantization rule. Black is 0, and white is 255, so the input image is a solid block of $108/255 \approx 42\%$ gray. Following the simple quantization rule above, output pixels in this halftone are black wherever the corresponding threshold is < 108, and white elsewhere. In the output halftone (fig. 1c), seven pixels are white. When the pixels are much smaller (fig. 1d), the eye will blend the black and white pixels into a perceived $7/16 \approx 44\%$ gray, close to the input tone.

Two properties of Bayer's threshold matrix (fig. 1b) determine the effectiveness of the halftoning illusion: the number of distinct thresholds, and the way they are arranged in the matrix. Consider first the number of distinct thresholds: all 16 numbers in the matrix are different. As a consequence, the output halftone can simulate 17 possible gray values.

To better understand this, suppose all pixels in the input image I have the same gray value. If all $I(x, y) = 0$, then after applying the simple quantization rule, all $H(x, y)$ will $= 0$ too: the output halftone is all black.

Now, if all $I(x, y) = 16$, then only the bottom-left pixel where $T(x, y) = 0$ will be white in the output halftone H ; all others will be black. This is shown in fig. 1e, in which I was a 12×12 image with all $I(x, y) = 16$. Since the input I is larger than the matrix T, T is repeated 3 times horizontally and vertically to cover the image. When the pixels are small the halftone H will be perceived as a block of gray with value 16/255 (fig. 1i).

Figure 1f shows the halftone when all $I(x, y) = 31$. Notice that two pixels in each tile block (where $T(x, y) = 0$ and 16) are white, yielding an apparent 32/255 gray.



a

108	108	108	108
108	108	108	108
108	108	108	108
108	108	108	108

b

240	112	208	80
48	176	16	144
192	64	224	96
0	128	32	160

Figure 1. a) An image I where all pixels are 108; b) Bayer's 4×4 threshold matrix (T); c) the halftone H obtained by applying T to I; d) the same halftone pattern, repeated with smaller pixels. In e) through h), T is applied to four different 12×12 images where all pixels are 15, 31, 47 and 63 respectively, yielding four halftones H which appear increasingly lighter from left to right; i) through l) show the same halftones with smaller pixels.

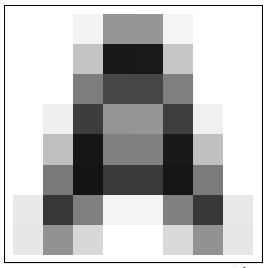
In fig. 1g, $I(x, y) = 47$ everywhere, and 3 pixels per block are white, while in fig. 1h, 4 pixels per block are white (yielding an apparent 48/255 gray). Clearly, if the input image is brighter, more pixels in the output halftone will be white, and a brighter gray will be perceived. Altogether, 17 possible gray levels can be simulated, because the threshold matrix has 16 different values.

Turning to the second property, consider the way the thresholds are arranged in the matrix $T(x, y)$. The threshold values are 0, 16, 32, 48, $\cdots$, 224, 240, but these consecutive values are not adjacent in the matrix: 0 and 16 are far apart, as are 16 and 32, 32 and 48, and so on. Such an arrangement makes this a *dispersed-dot* threshold matrix, and leads to output halftones without "clumps". This is indeed true for the halftones in figs. 1e through figs. 1h: the white pixels in these halftones are well separated, and do not clump together. Such clusters of pixels, be they black or white, tend to attract the eye's attention, and are artifacts that spoil the desired illusion of a single shade of gray.

In its most common application, printing, halftoning aims to be inconspicuous: a good dispersed-dot threshold matrix does not betray its presence, either by not producing enough gray tones, or by producing clumping artifacts. Nevertheless, because the threshold matrix must be repeated to cover the input image, a very small matrix may still be discernible, especially in smooth regions of the input image where the tone is relatively constant.

This paper presents a general way to make this repetition into a *feature.* If the threshold matrix encodes an interesting small picture M (a "motif"),

a



b

0	**0**	12	106	105	11	**0**	**0**
0	**0**	59	231	230	57	**0**	**0**
0	**0**	130	185	185	128	**0**	**0**
0	16	194	106	105	193	15	**0**
0	64	233	127	127	231	63	**0**
0	133	235	198	198	234	132	**0**
22	200	128	11	11	127	199	22
24	109	39	**0**	**0**	38	109	24

Figure 2. Decorative Halftoning: a) an 8×8 motif image; b) the corresponding naive threshold matrix T_N.

then repeating the matrix in a grid leads to an output *decorative* halftone: it shows the input image I, but also superimposes a grid of small motifs M on it.

2. Naive Decorative Halftoning

If the threshold matrix is built from the motif in a naive manner, the resulting halftones will have poor quality. Consider the 8×8 motif image M in fig. 2a, which shows the letter **A**, and the naive 8×8 threshold matrix T_N built from it, shown in fig. 2b. When this threshold matrix is applied to an input image, the output halftone will reproduce the input image's shades of gray (though not very well), with a repeated letter **A** superimposed in a grid.

A simple way to make the motif show through is to build the threshold matrix T_N by taking the negative of the image: if a pixel $p(x, y)$ in M has value v, the corresponding threshold is $T_N(x, y) = 255 - v$. As a result, bright pixels will have low thresholds, and dark pixels will have high thresholds (compare figs. 2a and b). With a matrix built this way, quantization will cause the motif to show through: For low thresholds $T_N(x, y)$, the test $I(x, y) > T_N(x, y)$ will succeed more often, tending to produce white output halftone pixels $H(x, y)$, corresponding to bright pixels in the motif. Conversely, high thresholds are likely to yield black output pixels, in places where the motif is dark.

Unfortunately, a naive threshold matrix built this way may have two undesirable properties:

1. Too few thresholds: its thresholds are not necessarily all different. For example, on the matrix in fig. 2b, the solid white background of the motif image leads to many corresponding thresholds with value zero. This is bad, because it limits the number of gray shades that the matrix can simulate in the output halftone.

2. Lack of dispersion: its thresholds do not necessarily occur in a dispersed-

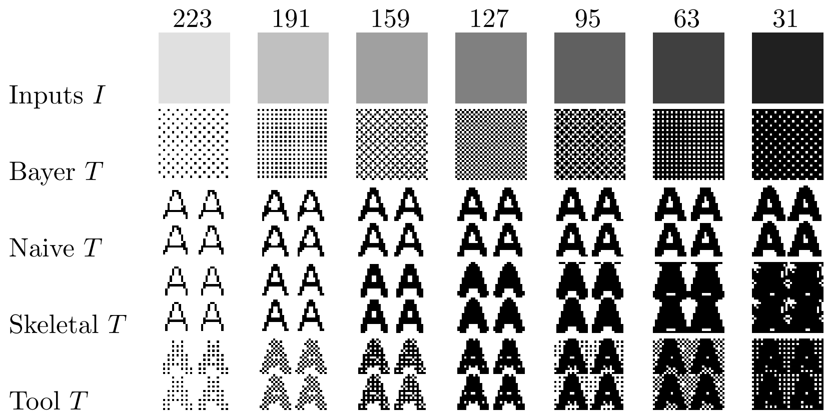


Figure 3. Comparison of decorative halftoning methods. Row by row: Several gray tones; Bayer's matrix; Naive decorative halftoning (note that darker tones are not reproduced); Ostromoukhov's method (the motif thickens to reproduce darker tones); The tool described here (the motif's shape is not changed).

dot pattern, unlike Bayer's matrices. In fig. 2b, thresholds with similar values are often adjacent. In fact, many similar thresholds may occupy large contiguous regions of the motif image (such as the background of the letter **A**, which is a contiguous region of zero thresholds). As a consequence, the corresponding regions in the output halftone may have undesirable clumps of white or black pixels.

The tool described here can overcome both these limitations. The solution is relatively simple, and can be deduced by comparing threshold matrices, shown in fig. 3. In this figure, the top row (Input) shows eight 32×32 solid blocks of gray. The second row (Bayer) shows halftones produced with Bayer's 16×16 dispersed-dot matrix. Since the input images are larger than the threshold matrix, the matrix is repeated as a square tile, twice both horizontally and vertically.

The next row (Naive) uses the simple decorative halftone method just described. Again, a 16×16 matrix is used, which must be repeated. Clearly, the darkest tones in the input image are not reproduced accurately: the halftones on the right side are too light. This is due to solid regions of zero thresholds in the motif's background. In those regions, $T(x, y) = 0$, so $I(x, y)$ always $\geq T(x, y)$, so the output halftone $H(x, y)$ is always white at those locations. Hence the perceived color is too light.

In the next row (Skeletal), Ostromoukhov's [2] method is used. This method works by first reducing the motif to a "skeleton" of polylines. The thickness of the lines is then varied to reproduce varying gray tones: darker halftones use thicker lines. The varying thickness is obtained by setting thresholds based

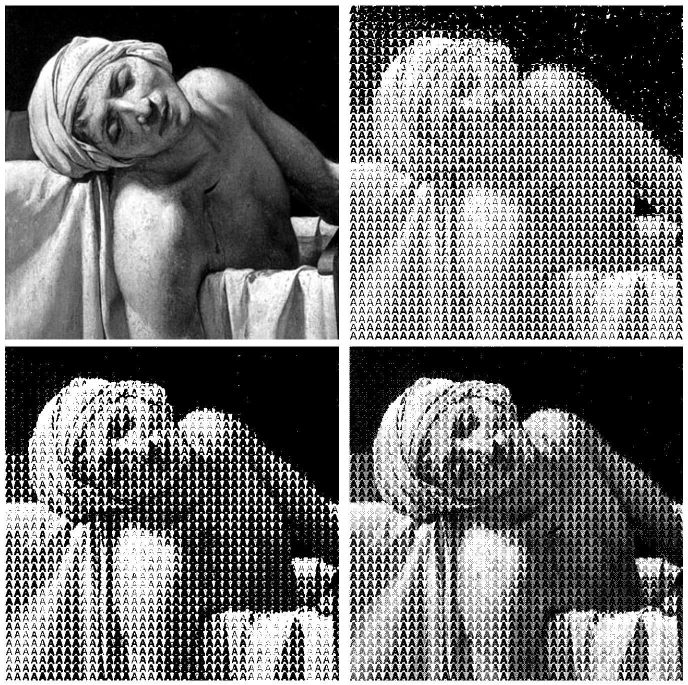


Figure 4. Row by row from top left: Detail from David’s “Death of Marat”; Naive decorative halftoning; Ostromoukhov’s method; The tool described here.

on distance from the skeleton: thresholds $T(x, y)$ decrease as the distance $d = |(x, y) - p|$ increases, where p is the point on the polyline closest to (x, y).

The bottom row (Tool) uses the tool described below. The shape of the motif is unchanged, but the background and foreground pixels in the motif seem to change gray tone. This is an illusion, since only black and white pixels occur in the halftone.

Figure 4 shows the three techniques applied to an image with very dark and very light regions. Notice that the naive method (top right) does not reproduce the some dark and light tones correctly. Ostromoukhov’s method (bottom left) gives better tone reproduction, but some spatial detail is lost in the mid-tone regions. The tool described here yields the image on the bottom right: of all the techniques, it preserves the most spatial detail, as seen in the figure’s face and body.

a

255	**125**	222	93	247	117	**214**	**85**
60	**190**	28	157	52	182	**20**	**149**
206	**77**	238	109	198	68	**230**	**101**
12	141	44	174	4	133	36	**166**
242	113	210	81	251	121	218	**89**
48	178	16	145	56	186	24	**153**
194	64	226	97	202	72	234	105
0	129	32	**162**	**8**	137	40	170

b

76	**36**	92	152	148	84	**64**	**24**
16	**56**	132	240	232	128	**8**	**40**
60	**20**	188	200	204	180	**68**	**32**
4	100	212	156	144	208	96	**52**
72	140	244	172	176	236	136	**28**
12	196	252	220	216	248	192	**44**
108	228	184	80	88	168	224	104
112	164	124	**48**	**0**	120	160	116

Figure 5. Building a better decorative threshold matrix: a) Bayer's 8×8 threshold matrix T_D; b) The threshold matrix T_M produced by this tool.

2.1. Method

The tool starts with a naive decorative threshold matrix T_N, and combines it with an existing dispersed-dot threshold matrix T_D of the same dimensions, producing a hybrid motif threshold matrix T_M. As we shall see, T_M is built by re-ordering subsets of thresholds in T_D, corresponding to the thresholds occuring in T_N.

A concrete example will help the explanation. Let T_N be the threshold matrix naively obtained from the 8×8 letter-**A** motif image in fig. 2a, and let T_D be Bayer's matrix of the same dimensions, shown in fig. 5a. Consider one threshold value found in T_N: zero. The subset of twenty zero thresholds in T_N make up the background of the motif, and are highlighted in fig. 2b. These thresholds are all the same, and are mostly adjacent in the matrix (are not dispersed), two properties that lead to poor tone reproduction, as discussed above.

On the other hand, the thresholds in Bayer's matrix T_D are all different, and are dispersed, which leads to good tone reproduction. To take advantage of this fact, consider the twenty zero thresholds in T_N, and let L_0 be the twenty thresholds in Bayer's matrix which correspond to those locations. The thresholds in $L_0 = [8, 12, 48, 60, \cdots, 230, 242, 255]$ are highlighted in fig. 5a. We want the thresholds in L_0 to have low values, thus tending to yield bright background pixels in the output halftone.

To achieve these low values while preserving the dispersed arrangement of thresholds, simply replace L_0 with the list $[0, 4, 8, \cdots, 72, 76]$: 8 becomes 0, 12 becomes 4, and so on. These new values are the lowest 20 thresholds in Bayer's matrix. The resulting changes are highlighted in fig. 5b.

This detailed example considered only one threshold value in T_N: zero. If the same process is applied to all the values occuring in T_N, then a threshold matrix is obtained which reproduces the motif image, but which has many different thresholds, arranged in dispersed positions. Algorithm 1 gives pseudocode for this process.

Algorithm 1. (Build decorative threshold matrix)

```
Input: a naive threshold matrix $T_N$
Input: a dispersed-dot threshold matrix $T_D$
Output: decorative threshold matrix $T_M$
$L \Leftarrow$ (empty list)
forall values $v$ in $T_N$ do
    $G_i \Leftarrow$ all pixels with value $= v$
    $L_i \Leftarrow$ (empty list)
    forall pixels $p$ in $G_i$ do
        $(x, y) \Leftarrow$ coordinates of $p$
        $t \Leftarrow$ threshold at $T_D(x, y)$
        add triplet $(t, x, y)$ to $L_i$
    end
    sort $L_i$ by increasing $t$
    append $L_i$ to $L$
end
forall $i$ from 0 to $|L| - 1$ do
    $(x, y) \Leftarrow$ coordinates of $L[i]$
    $T_M(x, y) \Leftarrow i * 255/(|L| - 1)$
end
```

3. Examples

Figure 6 shows two applications of the tool, both on the same input image (a detail from Michelangelo's fresco on the Sistine chapel celing). In fig. 6a, a single motif threshold matrix is used, repeated across the image. The motif image was a woolen texture. However, in fig. 6b, a different motif image is used for each tile position. Each motif image is a 16×16 picture of a letter of the alphabet. Both of these halftone outputs use a more complex quantization rule, which can handle more than two device colors. The rule is described elsewhere [3], but works with the same threshold matrices that have been derived for two-color halftones.

4. Advantages and Limitations

The main advantage of this tool is its flexibility: any motif image can be used, and there is no need to reduce motifs to linear skeletons. It can also work with other dither methods besides Bayer's: for example, Ulichney's void-and-cluster dispersed dither [8] could have been used instead for the matrix T_D in algorithm 1. Of course, this approach does have some limitations. The threshold matrix produced is dispersed, so it is not suitable for devices, like offset printing presses, which cannot produce tiny isolated dots of ink.

References

[1] K. Knowlton and L. Hammond, “Studies in Perception I”, 1966. Online at `translab.burundi.sk/code/vzx`.

[2] V. Ostromoukhov and R. D. Hersch, “Artistic Screening,” in *Proceedings of SIGGRAPH 95*, pp. 219-228, 1995.

[3] V. Ostromoukhov and R. D. Hersch, “Multi-color and artistic dithering,” in *Proceedings of SIGGRAPH 99*, pp. 425–432, 1999.

[4] B.E. Bayer, “An Optimum Method for Two-Level Rendition of Continuous-Tone Pictures”, IEEE 1973 International Conference on Communications, Vol. 1, June 1973, pp. 26-11 to 26-15.

[5] R. Floyd and L. Steinberg, “An adaptive algorithm for spatial gray-scale,” in *Proceedings Society Information Display*, **17**(2), pp. 75–78, 1976.

[6] D. L. Lau, G. R. Arce and N. C. Gallagher, “Green-Noise Digital Halftoning,” Proceedings of the IEEE, vol. 86, no. 12, pp. 2424-2442, Dec 1998.

[7] D. L. Lau and G. R. Arce, “Modern Digital Halftoning,” Marcel Dekker, New York NY, 2001.

[8] R. Ulichney, “The void-and-cluster method for generating dither arrays,” in *IS&T Symposium on Electronic Imaging Science & Technology*, (Proceedings of SPIE, Volume 1913), pp. 332–343, (San Jose, CA), 1993.

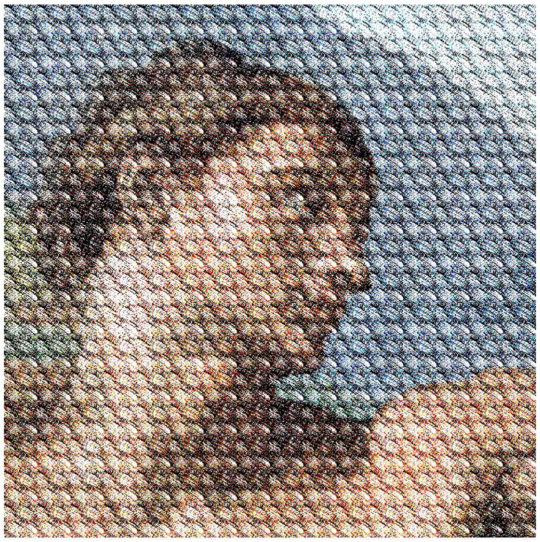
Web Information:

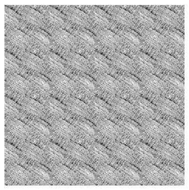
www.cs.unh.edu/∼ah/qdah

Alejo Hausner, University of New Hampshire, Durham, NH, 03824 USA (ah@cs.unh.edu)

Received [DATE]; accepted [DATE].

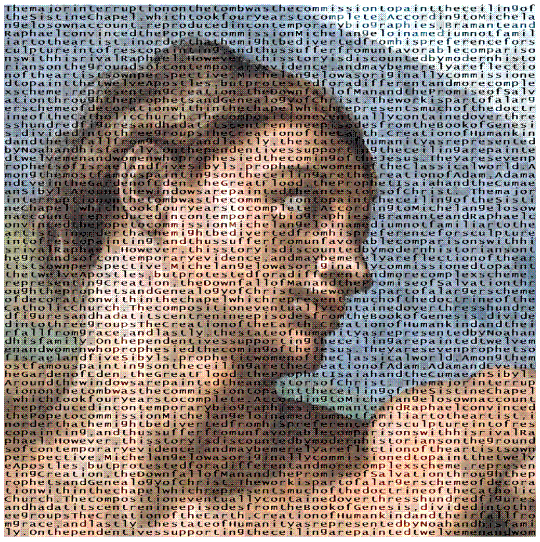
a

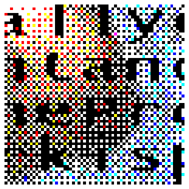




motif, repeated

b





detail

Figure 6. Decorative halftoning of Michelangelo's Adam: a) uses a fixed motif matrix (a knit texture), while b) uses a different black-and-white motif matrix on each tile; each motif is a letter of the alphabet.