EG UK Computer Graphics & Visual Computing (2017)
Tao Ruan Wan and Franck Vidal (Editors)

Colored AA Bitmaps

Abdalla G. M. Ahmed† Oliver Deussen

Department of Computer and Information Science, University of Konstanz, Germany

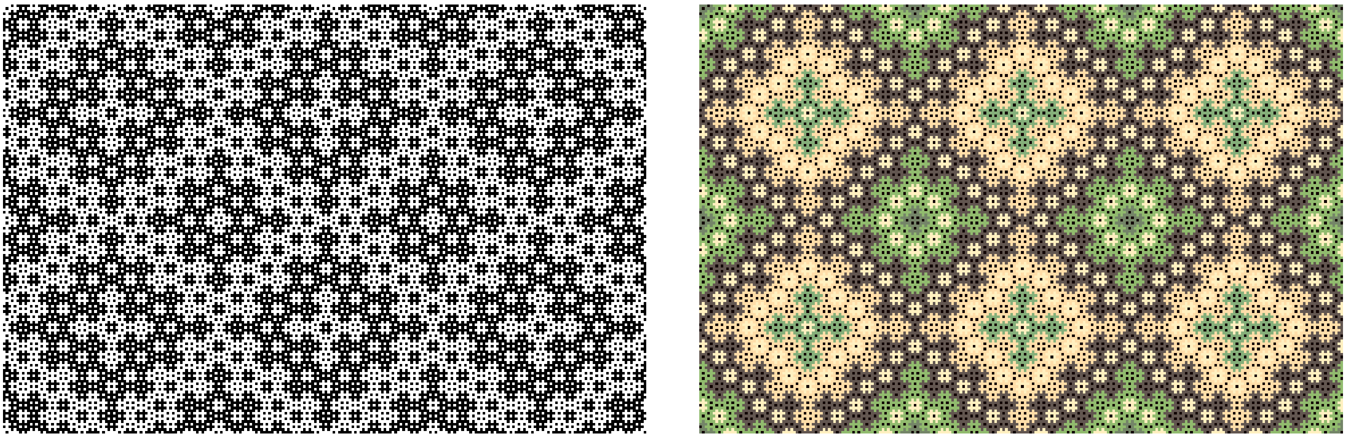


Figure 1: *An AA Bitmap (left) before, and (right) after colorization.*

Abstract

AA Bitmaps are ornamental monochrome bitmaps. We present an algorithm for colorizing a certain family of AA Bitmaps that relies on modular arithmetics.

1. Introduction

AA Bitmaps are ornamental monochrome bitmaps that are easy to construct from two binary strings. They comprise only four distinct columns, which makes them suitable for weaving design [Ahm13]. Figure 2 shows a few examples. The visual appeal and simple construction of AA Bitmaps make them a good candidate for aesthetic applications, but their monochrome nature drastically limits their utility. Thus, colorizing should substantially boost their value.

By examining the bitmaps in Figure 2 we may see that (c, d) are good candidates for colorizing: they comprise a few intertwined distinct elements, and we may assign to each element an entry in a color palette. In this paper we develop a method to colorize this family of AA Bitmaps, inspired by the colorizing algorithm of AA Patterns [Ahm11].

Algorithm 1 Drawing AA Outlines (reproduced from [Ahm13, Algorithm 1]).

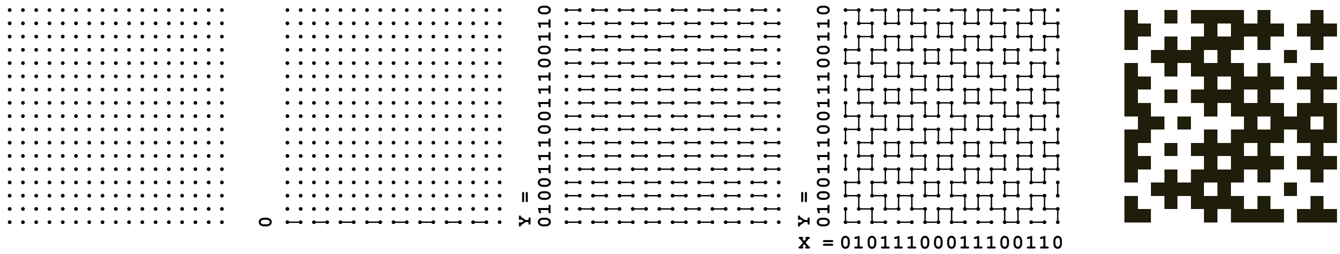
1. Start with a uniform 2D grid of points. See Figure 3(a).
2. On the first row join every other pair of points, creating a dashed line. See Figure 3(b). We label this row 0 if the first and second points are joined and 1 if the second and third points are joined.
3. Use a binary sequence **Y** to label each row 0 or 1 and create the appropriate dashed line. See Figure 3(c).
4. Similarly use a binary sequence **X** to label each column 0 or 1 and create the appropriate dashed line. See Figure 3(d).

2. Basic Definition of AA Bitmaps

AA Bitmaps are based on *AA Outlines* [Ahm12], which are piecewise contours that connect points in a regular grid, as explained in Algorithm 1 and illustrated in Figure 3. Each vertex in these outlines is degree 2, hence the whole contour is two-colorable. Since the outlines are based on a regular grid, the enclosed areas can

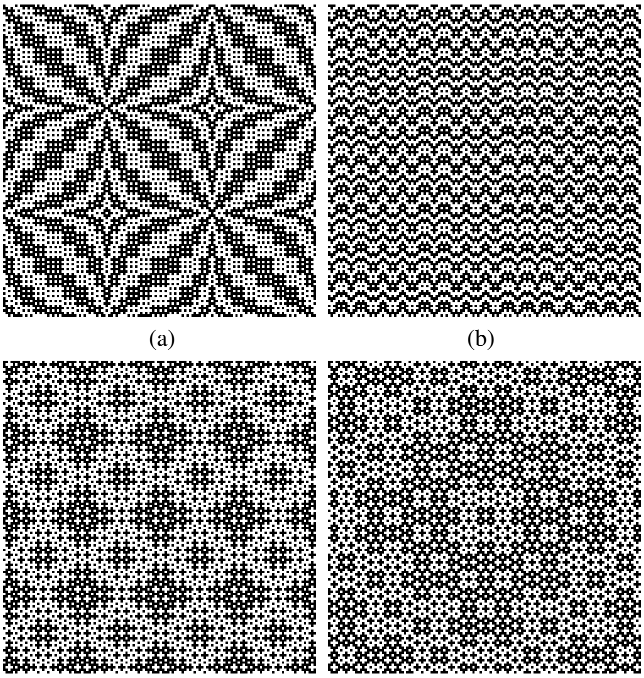
† Email:abdalla_gafar@hotmail.com

DOI: 10.2312/cgvc.20171281



(a) (b) (c) (d) (e)

Figure 3: *(a-d) Steps to build an AA Outline, and (e) the resulting AA Bitmap. Reproduced from [Ahm13, Figures 3 and 4]*



(a) (b) (c) (d)

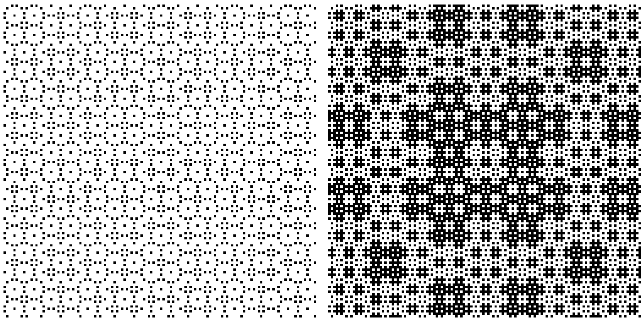
Figure 2: *Example AA Bitmaps. The ones in (c, d) are based on modular arithmetics.*

be viewed as pixels, and the two-colored outlines can be seen as bitmaps. AA Bitmaps is the name of such bitmaps.

Among all the possible $\{\mathbf{X}, \mathbf{Y}\}$ strings, we are interested in those obtained by modular arithmetics, using a pair $\{p, q\}$ of integers. Using C-language operators, the xth entry is defined as:

$$\mathbf{X}[x] = (p*x)\%(2*q) < q\ ?\ 0\ :\ 1\,. \qquad (1)$$

An analogous definition holds for $\mathbf{Y}$. We call the bitmaps obtained from such strings "modular AA Bitmaps". Ahmed [Ahm13] observed that these bitmaps are visually similar to AA Patterns. They are, however, much more efficient in utilizing the available spatial resolution; see Figure 4. Thus, colored AA Bitmaps should aesthetically outperform their AA Patterns counterparts since they admit more details.



(a) (b)

Figure 4: *A* 128×128 *(a) AA Pattern and (b) its AA Bitmap counterpart. Most of the pixels in the AA Pattern are wasted in the background.*

A modular AA Bitmap comprises intertwined *clusters*. A cluster is a contiguous region of pixels that can be traversed without crossing an edge in the outlines between pixels. Our goal is to paint similar clusters similarly; see Figure 1. Thus, the colorizing problem reduces to identifying distinct clusters.

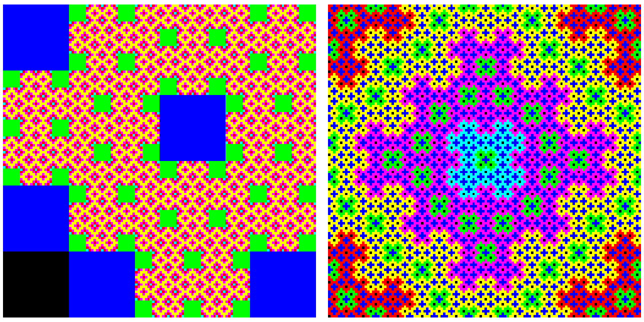
3. Index Space in Modular AA Bitmaps

We start by analyzing the structure of the bitmaps. The binary string $\mathbf{X}$ in Algorithm 1 defines the edges between the final pixels in the first row of the bitmap: '1' means that the current pixel and its western neighbor are in one cluster, '0' means they are not. All evenly-indexed rows resemble the first row, and the binary string is complemented in oddly-indexed rows. This complementation can be accounted for by adding a y component in Eq (1):

$$\mathbf{X}[x,y] = (p*x+q*y)\%(2*q) < q\ ?\ 0\ :\ 1\,.$$

Instead of thinking in terms of binary strings, we may now look at individual pixels. For each pixel (x,y) we compute a pair (u,v) of indices in $[0,2q)^2$:

$$\begin{array}{rcl} u & = & (p*x+q*y)\%(2*q) \\ v & = & (p*y+q*x)\%(2*q) \end{array}, \qquad (2)$$



(a) (b)

Figure 5: *(a) The index-space color map, and (b) the colorized pattern, for the AA Bitmap in Figure 2(d)*

which leads to this new definition:

> A modular AA Bitmaps is a partitioning of pixels into clusters such that for each pixel the western and southern neighbors belong to the same cluster iff $u < q$ and $v < q$, respectively.

4. Index-Space Coloring

Instead of tracing the clusters in the pixel space, we identify their index-space images, and use the index-space as a look-up color map. This way we avoid the problem of cropped clusters at the edges.

We start by allocating the $2q \times 2q$ map, and initialize it to a certain value interpreted as "free". We then find the next free index and trace it to identify the whole cluster. We maintain a list of indices in the current cluster, and recursively process it to find more neighbors. Processing an entry means checking the {north, east, south, west} neighbors to see whether they belong to the same cluster, and insert their indices in the list if they do. The indices of neighbors are obtained by adding $\{(q,p),(p,q),(q,2q-p),(2q-p,q)\}$, respectively, to the current index; all in modulo $2q$. Whether the neighbor belongs in the same cluster is determined by examining the index of the reference pixel for the western and southern neighbors, or the index of the neighbor for the northern and eastern neighbors. Each cluster is given a distinct number, and the belonging indices are marked with that number in the map to avoid adding the same index more than once.

The cluster identification process is repeated until no more free indices are left. Finally, similar clusters are identified by the number of pixels. In practice, this step can be combined with cluster identification in one pass. We provide a basic implementation in the supplementary material. Figure 5 shows an example result of the algorithm.

5. The Color Palette

Choosing a color palette is challenging: the clusters are intertwined in a complex way, and each color may come adjacent to all other colors. Luckily, even with coloring, generating AA Bitmap is sufficiently fast to work at an interactive speed, enabling the user to inspect a large number of patterns and develop his own insights to choose a color palette. We also found that using a color gradient over the palette entries usually produces appealing results. Figure 6 shows some results.

6. Conclusion

We successfully brought color to AA Bitmaps, enabling their use in many aesthetic applications, including textured backgrounds, picture frames, painted textiles, etc. The coloring algorithm is closely related to that for AA Patterns, but is simpler, and the results are visually richer by omitting the redundant background.

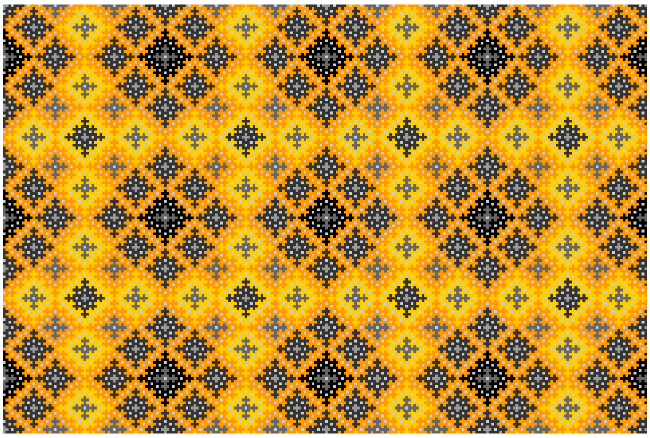
The primary challenging question for future research is to find an algorithm that can identify the nesting level, hence the color, of individual pixels by analyzing their coordinates; that is, without constructing the complete lookup table.

References

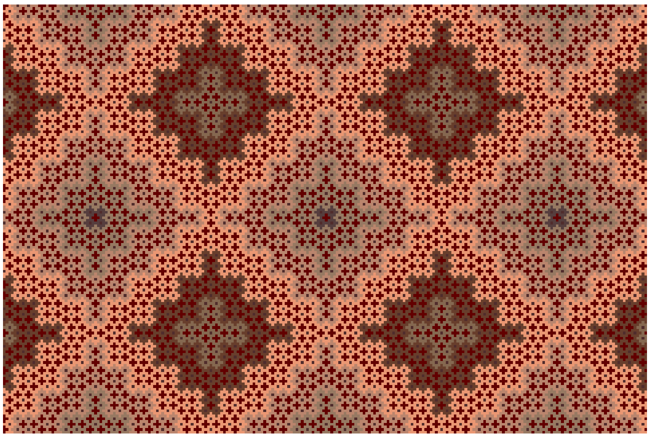
[Ahm11] AHMED A. G. M.: Pixel Patterns from Quantization Artifacts of Forward Affine Mapping. *Journal of Graphics, GPU, and Game Tools 15*, 2 (2011), 73–94. 1

[Ahm12] AHMED A. G. M.: On the Fractal Behaviour of AA Patterns. In *Theory and Practice of Computer Graphics* (Rutherford, United Kingdom, September 2012), Hamish Carr and Silvester Czanner, (Ed.), Eurographics Association, pp. 93–97. 1

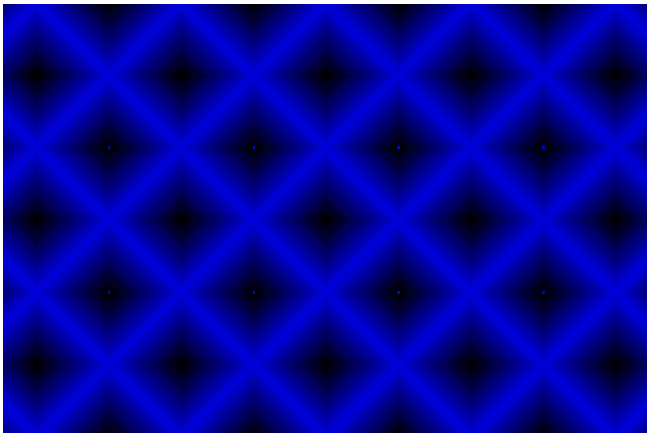
[Ahm13] AHMED A. G. M.: AA Weaving. In *Proceedings of Bridges 2013: Mathematics, Music, Art, Architecture, Culture* (Phoenix, Arizona, 2013), Hart G. W., Sarhangi R., (Eds.), Tessellations Publishing, pp. 263–270. 1, 2



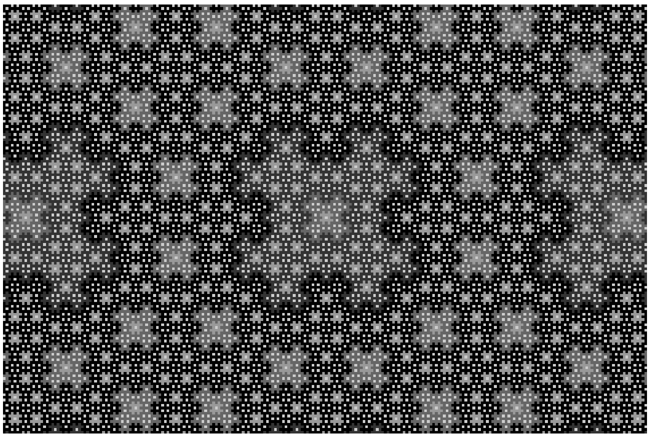
$q = 30, p = 11$



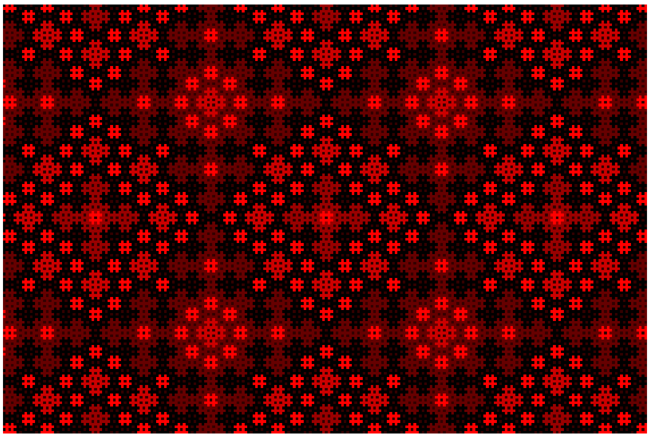
$q = 43, p = 24$



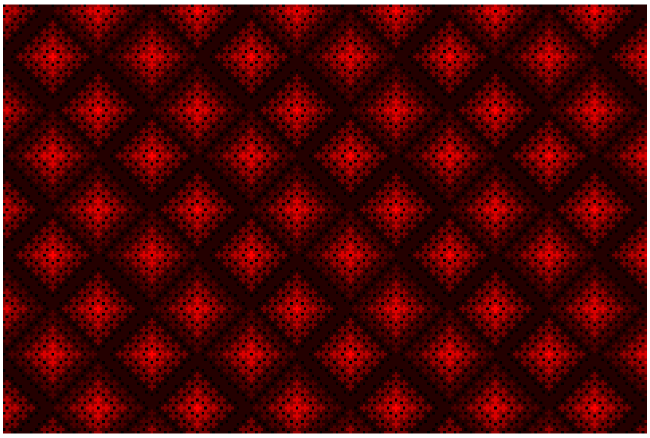
$q = 27, p = 26$



$q = 56, p = 15$



$q = 43, p = 12$



$q = 37, p = 13$

Figure 6: *Some results with different parameters and color palettes.*