Measuring Stipple Aesthetics in Hand-Drawn and Computer-Generated Images

Ross Maciejewski ■ *Purdue University*

Tobias Isenberg ■ *University of Groningen*

William M. Andrews ■ *Medical College of Georgia*

David S. Ebert ■ *Purdue University*

Mario Costa Sousa ■ *University of Calgary*

Wei Chen ■ *Zhejiang University*

When people compare a computer-generated illustration to a hand-drawn illustration of the same object, they usually perceive differences. This seems to indicate that the two kinds of images follow different aesthetic principles. To explore and explain these differences, the authors compare texture stippling in hand-drawn and computer-generated illustrations, using image-processing analysis techniques.

From hastily sketched figures on napkins to complex medical illustrations, hand-drawn images have long been a means of conveying information. Often, an artist condenses the information to the most important details, creating a simple, clear, and meaningful image. The artist accomplishes this refinement by directing attention to relevant features, simplifying complex features, or exposing obscured features. This selective inclusion of detail provides levels of expression often not found in photographs.

Computer graphics artists have adopted many traditional illustration techniques in nonphotorealistic rendering (NPR). They have particularly focused on traditional pen-and-ink techniques, attempting to mimic artists' strokes, textures, and tones through the placement of lines and points of varying thickness and density in computer-generated images. In this article, to highlight the differences between computer-generated images and hand-drawn images, we will focus solely on *stippling*, a pen-and-ink subset. Figure 1 shows examples of stippling in hand-drawn illustrations.

In stippling, the artist places dots on a surface of contrasting color to obtain subtle shifts in value. These dots can vary in size, volume, and arrangement to create the illusion of different texture, tone, and shape. To visually describe forms and objects, the artist begins by placing dots randomly on the surface and then gradually fills in areas from these seed dots.[1] Thus, stippling represents fine details and textures using points. Since points are the simplest graphic primitives, automating stippling is an attractive goal. Researchers have developed many techniques for creating interactive, computer-generated stipple renderings.[2,3] However, an important question has gone unanswered: Do computer-generated stipple images have the same aesthetics as hand-drawn stipple images?

To answer this question, we use image-processing texture analysis techniques. The first step in texture analysis is defining the concept of texture. Image processing uses two main approaches to defining texture: structural and statistical. The structural approach defines texture as a set of primitive texels that contain a regular or repeated relation-

 Published by the IEEE Computer Society

ship; the statistical approach defines texture as a quantitative measure of the arrangement of intensities in a region. Examples of structural textures are wood grain and brush strokes; statistical textures include random patterns such as those of stone or ice. To apply common texture analysis methods to NPR techniques, we must first determine which type of texture the technique produces: structural or statistical.

A cursory glance at a stipple image makes it clear that structural analysis is not appropriate because isolating a pattern for analysis would be difficult, if such a pattern even exists. Moreover, most stippling systems rely on a random distribution of stipple points. Therefore, statistical texture analysis is the most appropriate approach for analyzing stipple textures, and structural analysis methods found in recent texture synthesis algorithms (for example, in Wang and colleagues[4]) are not appropriate. To analyze the aesthetics of hand-drawn and computer-generated stipple textures, as well as comparable photographs of natural textures, we use image analysis metrics associated with the statistical texture approach—specifically, the gray-level co-occurrence matrix.[5] Although we could use other statistical methods (for example, in Qin and Yang[6]), we have found GLCM most appropriate for analyzing stipple textures.

From the GLCM, we calculate texture properties such as energy, correlation, and contrast. By comparing these properties in an array of stipple images, we can learn which features these textures have in common and where current stipple algorithms fail in producing textures similar to hand-drawn ones.

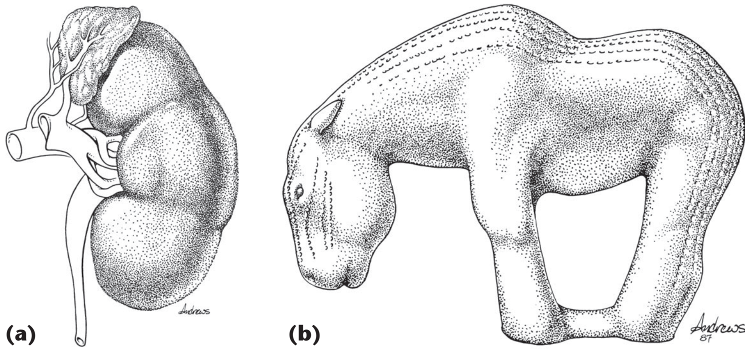


Figure 1. Stippling in hand-drawn scientific illustrations, depicting tone and shape features.

Computer stippling systems

Many computer illustration systems incorporate stippling algorithms. Deussen and colleagues were among the first researchers to computationally create stipple images.[2] They render polygonal models into a continuous tone image and then convert these target images into a stipple representation. To do this, they apply half-toning techniques to arrive at an initial stipple distribution and then interactively apply relaxation based on centroidal Voronoi diagrams (Lloyd's algorithm), using specialized brushes to space the stipple points more evenly. They suggest using a Poisson disc distribution to simulate the artistic stipple distribution.

In contrast to this interactive approach, Secord uses a fast probabilistic method that places small stipple primitives.[3] He also performs iterative relaxation using centroidal Voronoi diagrams but doesn't require interactive adjustment of the stipple spacing with brushes. Instead, he weights the computation of the centroidal Voronoi diagram on the basis of the input image's local gray level. As a result, stipples are automatically packed more densely in dark regions and more sparsely in lighter regions.

Schlechtweg, Germer, and Strothotte (hereafter referred to as Schlechtweg) take a different general approach.[7] They have created a multiagent system called RenderBots to position NPR strokes, including stipples, based on a stack of geometric buffers generated from a 3D model. Each stipple bot represents a single stipple dot and sees only its local neighborhood. It is capable of simple actions such as trying to go to dark regions while avoiding other stipple bots. By simulating this behavior, RenderBots achieves a similar stippling distribution as the previously mentioned techniques, albeit with more randomness. Sousa and colleagues (hereafter referred to as Sousa) approximate stippling by using short, serrated ink strokes modeled directly over the mesh's edge.[8] This technique places and stylizes strokes using parameters extracted from the mesh's surface, resulting in the precise-ink illustration style.

Two stippling aesthetics

Although NPR stipple techniques capture many aspects of hand-drawn image styles, there are visible dissimilarities between computer-generated and hand-drawn images. In a recent study in which participants compared hand-drawn and computer-generated pen-and-ink drawings, Isenberg and colleagues found that participants usually could distinguish between the two categories.[9] The distinguishing features of stipple images were stipple point density, the use of shading, and the presence of artifacts. Artifacts included unwanted regularity in computer-guided dot placement, leading to the formation of lines, in contrast to the more random placement of dots in the hand-drawn images. There were also intentional artifacts in the shape of hand-placed dots, in contrast to the very regular, rounded computer-generated dots. These differences might serve a purpose beyond added visual interest: an analysis of texture properties of hand-drawn and computer-generated images could potentially be used to classify them into two separate categories if the differences are common within a set.

Isenberg and colleagues' study also showed that the differences between hand-drawn and computer-generated images did not necessarily lead people to appreciate or like one category more than the other. On the contrary, participants said that they liked both categories of images, each for different reasons, and would use them in different domains. Thus, we can conclude that although NPR techniques have potential for improvement, hand-drawn and computer-generated stippling involve two different aesthetics.

Although no formal metric has been introduced for distinguishing hand-drawn and computer-generated stippling, differences clearly exist.

Hand-drawn stippling aesthetics

As a visual style, stippling fills an important role in medical, scientific, and technical illustration. With its ability to depict tonal variations, stippling is well suited for illustrating objects with fine detail, subtle texture, and small changes in shading. A limitation of line illustration techniques in general is that the individual marks must be smaller than the finest detail to be depicted. In stippling, this is rarely a concern. Gradients and soft edges are relatively easy to create by varying the size and density of marks. However, long lines and hard edges are relatively difficult to create using stipples. For particular illustrations, stippling is the preferred choice because other pen-and-ink techniques, such as hatching, might be mistaken for contours in the images. The random placement of stipples in medical illustrations provides for tone and shape, while not creating any undesired directional cues. This is not to say, however, that creating structures in stippling is always undesired.

In creating stippled illustrations, artists must address several interrelated issues. First, they must choose the physical characteristics of the marks, including size, variability, and frequency. Second, they must choose edge- and shape-handling techniques, which affect shape and form recognition as well as depth cues from interacting shapes. Finally, they must choose form-shading techniques, which emphasize and deemphasize objects. All these factors contribute to the perceived aesthetics of hand-drawn stipples.

NPR stippling aesthetics

Computer artists typically create stipples in NPR by placing points explicitly or using small short strokes that approximate stippling. NPR stipple creation involves choosing a stipple primitive and a stipple distribution. Algorithmic stipple placement enables computer-generated stipple illustrations to use a far greater number and thus a higher density of stipple points. This means that smaller dots can be used, resulting in potentially finer detail. Strict use of a model and a shading computation leads to an almost realistic depiction of the illustrated shapes.

Another factor influencing the aesthetics of NPR stippling is the choice of dot or line shapes. Explicit point placement techniques usually employ dots ranging from perfectly symmetric to slightly irregular, asymmetric, but still rounded marks to simulate hand-drawn dot-by-dot stippling.[2,3] Short strokes are typically asymmetric to replicate the precise-ink stippling technique.[8] Depending on the type of rendering, the mark size is sometimes close to the final resolution of the pixel image, leading to pixels or small groups of pixels representing one dot or one short stroke. Again, all these factors contribute to the perceived stipple aesthetics.

Distinguishing the aesthetics

Although no formal metric has been introduced for distinguishing hand-drawn and computer-generated stippling, differences clearly exist. One difference might be the potential preciseness of computer-generated stipples compared with hand-drawn stipples, as discussed by Isenberg and colleagues.[9] This preciseness could lead viewers to sense sterility or rigidness in computer-generated stipples. Such qualities are not always undesirable. More detailed structures can show more accurate shape, shading, and illumination. Furthermore, computers are very good at creating patterns, which can enhance the perception of object features. Such structures might be more difficult to represent in hand-drawn images. In contrast, hand-drawn images can seem less sterile to viewers because many natural surfaces have statistical properties that imply self-similarity, meaning that any extracted sample will have properties similar to the whole. Self-similarity is often found in natural textures. If these natural properties exist in hand-drawn images, but not in computer-generated images, this difference could explain the visible dissimilarities between hand-drawn and computer-generated images. However, if we consider this in terms of textures, it is possible that placing points explicitly, as opposed to placing small marks, will fall into separate texture classes (that is, statistical or structural).

Statistical texture analysis methods

As we previously stated, the two common texture analysis methods are structural and statistical. A structural approach works well for regular patterns, but the lack of discernible patterns in stippling requires using a statistical approach. There are many statistical texture analysis algorithms designed to represent textures for comparison. GLCM, Fourier power spectra, and texture spectra are some of the more common approaches.

We have chosen the GLCM algorithm to analyze stipple textures for several reasons. First, studies in perceptual psychology have shown that the GLCM closely matches levels of human perception.[10] Second, many studies have shown that this method outperforms others in texture discrimination. For example, Weszka, Dyer, and Rosenfeld analyzed GLCM performance in comparison with three other algorithms on a set of aerial-photographic terrain samples.[11] Their study showed that GLCM outperformed other algorithms with respect to textural feature derivation. Finally, we use the GLCM algorithm rather than more recent algorithms, such as advanced implementations of the gray-level difference histograms[12] and gray-level aura matrices.[6] Although these methods are appropriate for texture analysis and synthesis, we don't need their ability to catch structures in textures because stipple artists avoid producing oriented textures or unintended patterns.[1] Thus, algorithms that measure anisotropy and texture symmetry are not necessary and might not be adequate for comparing stipple textures.

Applying GLCM to stipple textures

A GLCM is a two-dimensional array L in which rows (r) and columns (c) represent a set of possible gray-tone values G. The value $L(i, j)$ indicates how many times value i co-occurs with value j in a given spatial relationship defined by $\mathbf{d}$. If we allow $\mathbf{d}$ to be a displacement vector (dr, dc), where dr is the displacement in rows and dc is the displacement in columns, the co-occurrence matrix $L_\mathbf{d}$ for an image, I, is defined

$$L_\mathbf{d} = |\{[r, c]|\ I[r, c] = i \text{ and } I[r + dr, c + dc] = j\}|$$

Figure 2 illustrates this equation with a simple 4 × 4 image I and two different co-occurrence matrices for I: $L[0, 1]$ and $L[1, 1]$. In this example, I consists of three different gray levels, denoted 0, 1, and 2. For $L[0, 1]$, we have used a horizontal

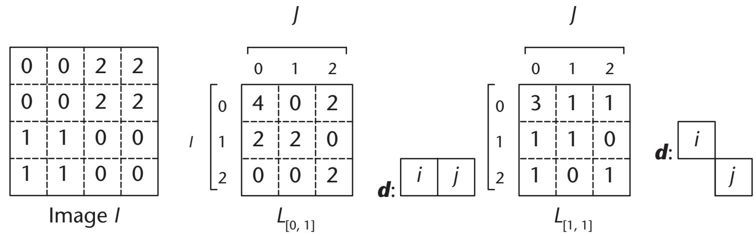


Figure 2. Two GLCMs (L[0, 1] and L[1, 1]) for sample image I, with three levels of gray values (0, 1, 2). The GLCMs are indexed by the gray values, and each matrix entry contains the co-occurrence of value i with value j in spatial relationship d.

displacement vector of unit length, and position [0, 0] in the GLCM has a value of 4, indicating that $j = 0$ appears directly to the right of $i = 0$ four times in I. Similarly, in $L[0, 1]$ position [0, 1] has a value of 0, indicating that $j = 1$ never appears directly to the right of $i = 0$ in I. If we use a diagonal displacement vector of unit length and direction (1, 1), our GLCM for I will be $L[1, 1]$. Here, in $L[1, 1]$, position [0, 0] has a value of 3, indicating that $j = 0$ appears directly diagonal to $i = 0$ three times in the image.

Once the GLCM, $L_\mathbf{d}$, is calculated, it is typically normalized to a matrix, $N_\mathbf{d}$, so that the values lie between 0 and 1. Thus, we can regard the co-occurrence values of $N_\mathbf{d}$ as the probability that one gray level will occur with respect to another gray level in direction $\mathbf{d}$:

$$N_\mathbf{d} = \frac{L_\mathbf{d}[i, j]}{\sum_i \sum_j L_\mathbf{d}[i, j]}$$

From this result, we can see that the GLCM captures texture properties but is not directly useful for further analysis, such as comparing two textures. Instead, we compute textural statistics from the GLCM to represent the texture more compactly. Of the 14 textural statistics proposed by Haralick, Shanmugam, and Dinstein,[5] Baraldi and Parmiggiani found six independent texture measures: energy, contrast, variance, correlation, entropy, and inverse difference movement. Baraldi and Parmiggiani investigated the meanings of these statistical GLCM parameters.[13] They found energy and contrast the most efficient parameters for discriminating different textural patterns. Furthermore, they found that in an image without probable linear dependencies, correlation is efficient in statistically discriminating areas with low textural content. Given the random placement of the initial stipple distribution by artists and computer programs alike, we can assume that the probability of linear dependencies in a stipple texture is low. Thus, our work uses only three texture statistics for comparison: contrast, energy, and correlation. We derive these features from the normalized co-occurrence matrix.

The first property we analyze is texture contrast. Contrast represents the difference between the highest and lowest values of a contiguous set of pixels. This implies that a low-contrast image is not characterized by low gray levels but rather by low spatial frequencies. Thus, GLCM contrast is highly correlated with spatial frequencies. We define contrast as

$$Contrast = \sum_i \sum_j (i-j)^2 N_{\mathbf{d}}[i,j]$$

The next property we analyze is energy. Energy measures textural uniformity. When only similar gray-level pixels are present in an image patch, a few GLCM elements will be close to 1, while many elements will be close to 0. In that case, energy reaches values close to its maximum. Therefore, high energy values occur when the texture's gray-level distribution has either a constant or a periodic form. We define energy as

$$Energy = \sum_i \sum_j N_{\mathbf{d}}^2[i,j]$$

Finally, we look at texture correlation. GLCM correlation is expressed as the correlation coefficient between two gray levels in the texture. High correlation values (near 1) imply a linear relationship between the gray levels of a pair of pixels. Correlation can be measured either in high- or low-energy situations and is uncorrelated with the GLCM contrast metric. We define correlation as

$$Correlation = \frac{\sum_i \sum_j (i-\mu_i)(j-\mu_j) N_{\mathbf{d}}[i,j]}{\sigma_i \sigma_j}$$

where μ_i and μ_j are the means and σ_i and σ_j are the standard deviations of the row and column sums of $N_{\mathbf{d}}$.

To effectively compare different stipple textures, we gathered a set of computer-generated images representing systems that use 3D geometry, producing vector graphic output or pixel images.[3,7,8] In addition, four medical and scientific illustrators provided hand-drawn images for our study: William M. Andrews, Andrew Swift, Emily S. Damstra, and Gerald P. Hodge. We compared the computer-generated images with the images of one or more of the artists. We also compared two hand-drawn images with two natural-texture photographs.

To compare the texture samples shown in Figure 3a and other figures in this article, we performed several image preprocessing steps. We scan-converted and down-sampled each vector image so that the smallest stipple covered approximately one pixel in each image, thus letting us apply GLCM to analyze stipple density. Similarly, we scanned the hand-drawn images in black and white at high resolution and down-sampled them to approximately the same resolution as the vector images. After down-sampling all the images, we cropped a small texture sample of 50 × 50 pixels from each image. Next, we performed an initial analysis of these texture samples to determine the optimal parameters needed to create the GLCM.

To use the GLCM effectively for texture analysis, we must first make reasonable choices of GLCM parameters. Texture analysis requires choosing window size, offset direction, offset distance, number of gray levels to quantify the picture, and statistical measures. We have already explained the texture measures we use—contrast, energy, and correlation. Furthermore, given the randomness of stipple patterns and the fact that stipple images are essentially black and white, we planned to create texture statistics along only one directional vector and quantize the image to a small number of gray levels.

The purpose of our first analysis, therefore, was to make sure that the offset direction would not affect our texture statistics. The relation between the reference pixel and its neighbor can be in any one of eight directions (north, south, east, west, or the four diagonals). Since north is opposite of south, and likewise east of west, we can reduce the number of directions to be analyzed to four. We computed texture statistics using four offset directions: northeast, east, southeast, and south. We calculated 40 different GLCMs for each offset direction, with magnitudes of 1, 2, …, 40, where we used the maximum magnitude 40 to avoid problems near the edges of the texture. We used 12 texture samples, spanning our range of hand-drawn and computer-generated images. We then plotted texture statistics, and the results of this analysis confirmed that the offset direction does not affect the texture statistics for stipple textures. This analysis ensures that the stipple patterns lack any consistent directionality.

In addition to ensuring the independence of texture statistics from the offset direction, we had to analyze the effects of down-sampling and scanning. In our stipple images, only two unique gray levels should represent black and white. To verify that scanning and down-sampling didn't introduce any graying artifacts, we calculated the GLCM multiple times by changing the number of gray levels used to quantify an image. We com-

Courtesy of Isenberg et al.[9]; copyright respective artists

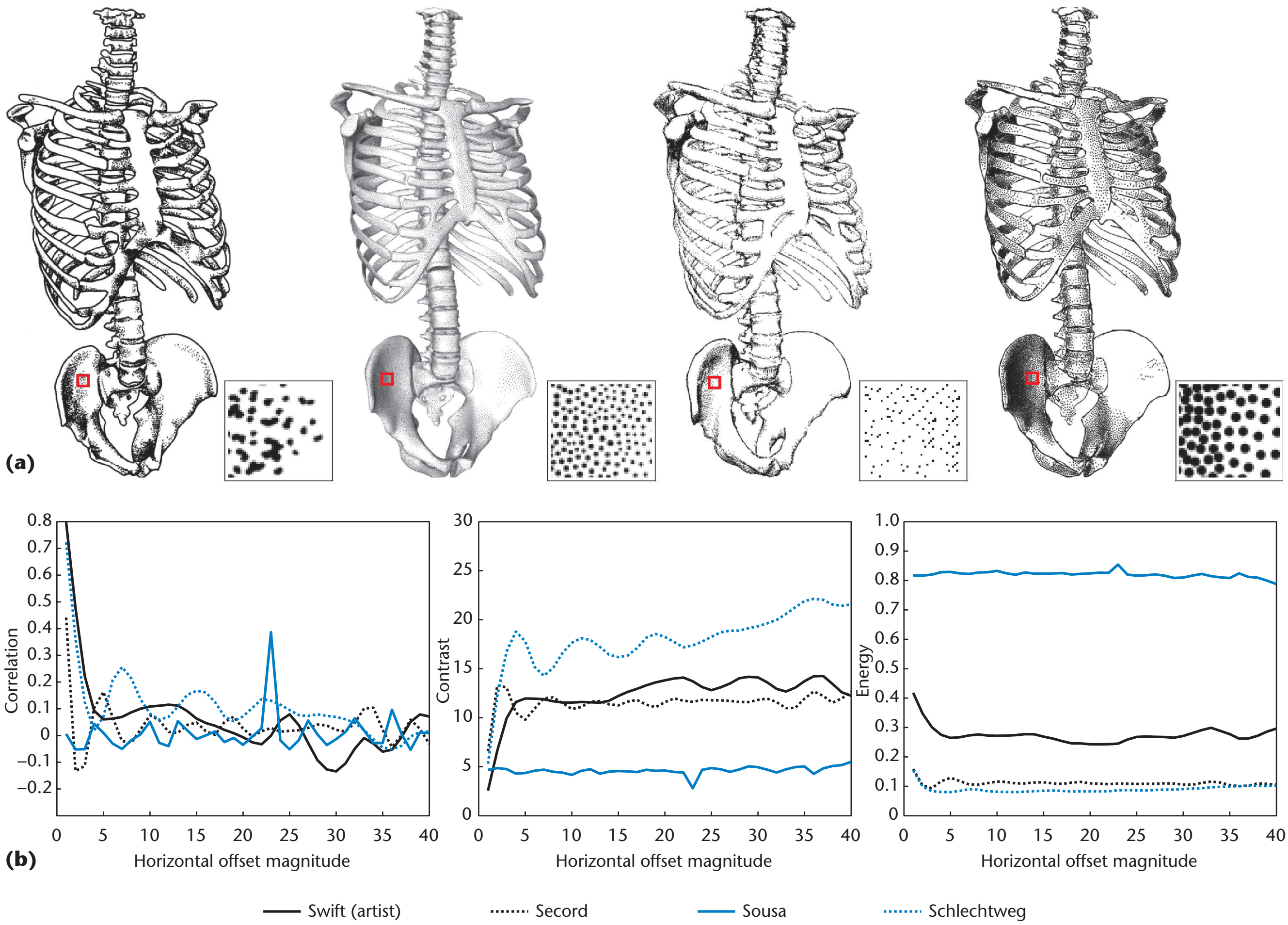


Figure 3. Hand-drawn and computer-generated stippling in anatomical illustrations. (a) Stipple images of a partial skeleton, with detail samples indicated by red squares. Images from left are a hand-drawn illustration by Andrew Swift and illustrations generated with techniques of Secord,[3] Sousa,[8] and Schlechtweg.[7] (b) Texture statistics from samples in (a): correlation (left), contrast (middle), energy (right).

puted texture statistics using the same texture samples as the directional analysis and varying the gray levels rather than the direction. We used gray-level values of 2, 8, 64, and 128 for quantization. We used a horizontal offset vector and calculated 40 different GLCMs for each quantization. We then plotted texture statistics. The results confirmed that the number of gray levels used for quantization doesn't affect the energy or correlation statistics for stipple textures, implying that only two different gray levels exist in the images. Contrast, on the other hand, varied widely because this property is scaled by the number of gray levels; however, results remained consistent within the scaling.

From our earlier analysis, we chose to create GLCMs for texture analysis using 50×50-pixel texture samples. We then created GLCMs from these samples, using horizontal offsets of (0, 1), (0, 2), …, (0, 40). Each GLCM uses eight different gray levels to quantize the image. Then we calculated and compared correlation, energy, and contrast statistics for the textures.

The sample size choice was necessary to avoid taking unwanted lines or other structures from the images under analysis. However, a sample of any size can be used as long as it contains only the stipple texture. The results remain consistent even when the sample size varies.

Results

Figure 3b presents the three texture statistics plots of the texture samples in Figure 3a. Each plot displays the corresponding texture statistic in relation to the magnitude of the offset direction. Figure 3b (left) shows the texture correlation of the four samples. Three of the correlation plots (Swift's, Secord's, and Schlechtweg's) have smooth curves falling from a high correlation at an offset of one pixel to a low correlation as the offset approaches approximately five pixels. In contrast, Sousa's texture correlation plot resembles that of a random distribution. The smooth fall-off of Schlechtweg's and Secord's plots is the only characteristic similar to Swift's correlation plot. Schlechtweg's plot contains an oscillating function, indicating an underlying pattern in the texture structure. Secord's plot shows similar oscillations along with a sharp secondary spike, also not present in the artist's plot.

Figure 3b (middle) shows the texture contrast of the same four samples in relation to the offset magnitude. Again, we see smooth curves in Schlechtweg's and Swift's graphs. These curves indicate that the contrast to neighboring pixels is low, but farther from the reference pixel, the contrast becomes high. Schlechtweg's graph again displays periodicity, indicating an underlying pattern in the texture. Secord's plot also indicates a low contrast to neighboring pixels; however, the change in contrast levels in this image is noticeably sharper than in Schlectweg's image or the hand-drawn image (Swift). Again, Sousa's plot indicates a random distribution of pixels with a varying contrast among pixels across the board.

Figure 3b (right) shows the samples' texture energy in relation to the offset magnitude. Again, we see discrepancies among the techniques. The hand-drawn image has a smooth curve, with a slightly higher set of energy values corresponding to pixels neighboring the reference. This value levels off to a constant value, indicating that once the offset is a certain distance (about five pixels) from the reference, the texture maintains a constant level of uniformity, where a value of 1.0 for energy would indicate a constant image. Schlechtweg's plot has a curve similar to Swift's, simply offset at a lower value. Secord's plot has a similar but less pronounced shape, with a value near zero, indicating that the uniformity of the texture produced is very low. Similarly, Sousa's plot is a constant value with a fairly high energy level, indicating a random stipple distribution. The high energy most likely results from a low stipple distribution over the image.

These plots show that hand-drawn textures tend to have a strong local correlation among neighboring pixels, indicating that hand-drawn stipple points tend to have other stipple points drawn nearby, and the distribution tapers off as the artist works outward from that area. This strong local correlation also accounts for the low contrast among neighboring pixels and the uniform energy level. In contrast, the computer-generated textures either show no local correlation (Sousa) or contain unwanted patterns (periodicity for Secord and Schlechtweg).

We analyzed another set of images from the same systems and artist. Figure 4a shows the four stipple images. Again, we took texture samples from each image, analyzed them with the GLCM, and plotted the same texture statistics. Comparing the graphs in Figure 4b with those in Figure 3b shows that the textures from the stippled arrowheads have similar texture statistics to those of the stippled skeletons. The main difference in Figure 4 is in Sousa's texture, where the contrast and energy values have changed because the sample came from a densely stippled rather than a sparsely stippled area. To strengthen our findings, we took a second sample set from these images and applied the GLCM analysis. The results (Figure 4c) are comparable to those in Figure 4b.

Our next analysis used images and texture samples from the same three computer-stippling systems, as well as hand-drawn images by William M. Andrews and Andrew Swift. Figure 5a (page 70) shows the images and samples. Figures 5b-5d show texture statistics for the samples. Here, the two hand-drawn textures have similar texture statistics, although the two samples exhibit different tones. When we compare the texture statistics in Figures 5b-5d with our previous results, we see similar features among the different systems.

Next, to demonstrate the lack of directionality in stipple textures, we applied the GLCM algorithm to the texture samples of Figure 5a, varying the offset vector direction. Figure 5b shows texture statistics based on a horizontal offset. In Figure 5c, the statistics are based on a vertical offset. Finally, Figure 5d shows the statistics based on a southeast offset. In each case, changing the directional vector had little effect on the textural descriptors, so we conclude that these textures are directionally invariant. We applied other directional vectors to these textures as well as to other texture samples, and all results were consistent with the findings presented here.

Finally, we compared two hand-drawn images (a cicada by Gerald P. Hodge and an artifact by Emily S. Damstra) with two natural-texture photographs. Figure 6a (page 71) shows the images and samples. The two natural textures (granite and quartzite) appear to have a random surface pattern, making them appropriate for GLCM analysis. Figure 6b shows the texture statistics for these samples. Although the curves representing the hand-drawn images aren't quite as smooth, they are still similar to those of the previous hand-drawn textures. The natural textures show a strong local coherency in neighboring pixels with a much more pronounced fall-off curvature than even the hand-drawn textures. Likewise, the low local-contrast level and the low energy values of the real-world textures are much closer to the statistics of hand-drawn textures than to those of computer-generated textures.

Courtesy of Isenberg et al.[9]; copyright respective artists

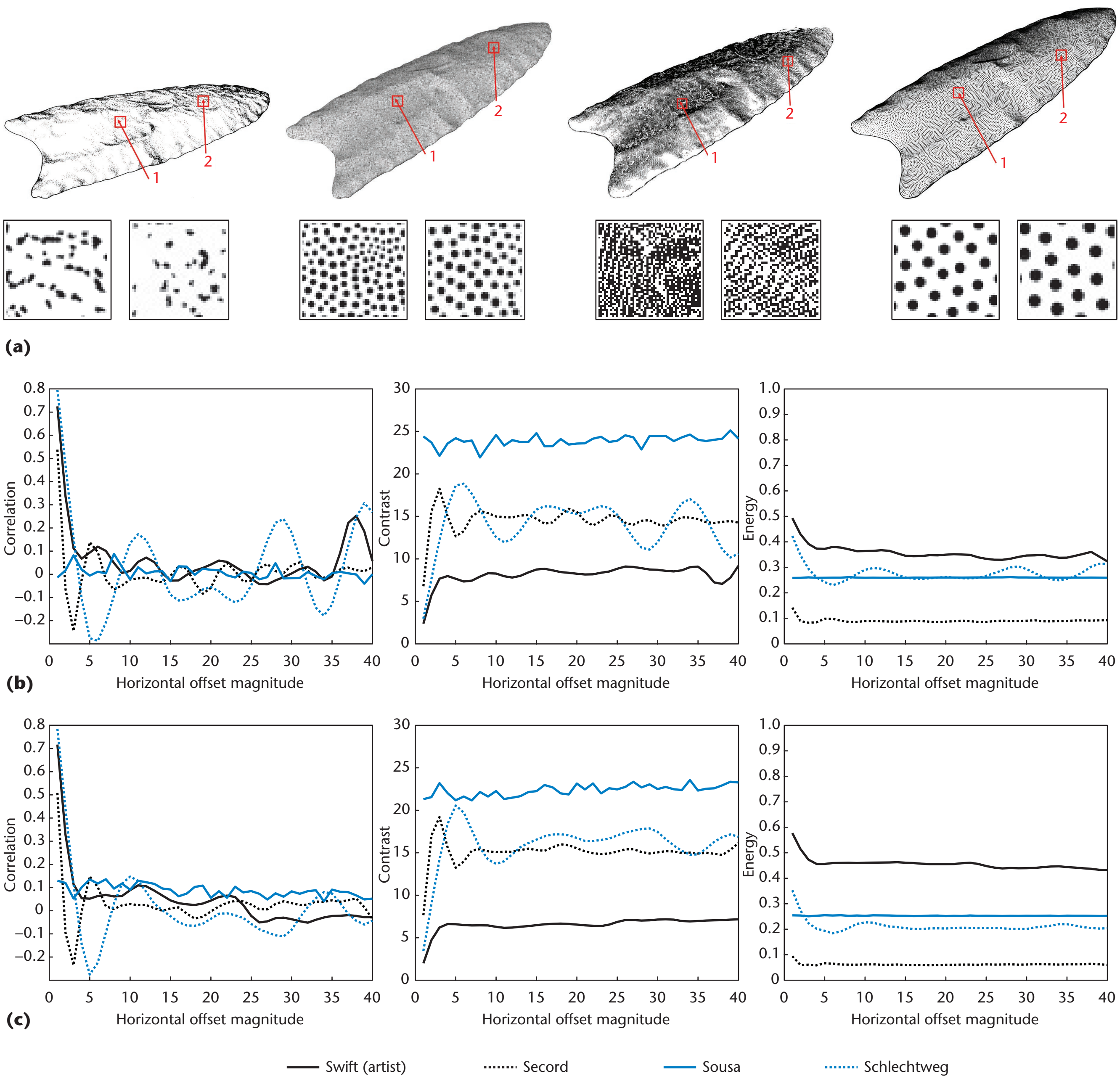


Figure 4. Hand-drawn and computer-generated stippling in archaeological illustrations, with two samples each. (a) Stipple images of an arrowhead, with enlarged images of stipple samples. Images from left, are a hand-drawing by Swift and computer-generated illustrations from Secord, Sousa, and Schlechtweg. (b) Texture statistics from sample set 1 and (c) texture statistics from sample set 2.

To further verify these results, we applied the GLCM algorithm to a larger texture sample in our images. Figure 7a (page 72) shows one example. Here, each sample is approximately 150 × 150 pixels and is down-sampled so that the smallest stipple covers one pixel. We created GLCMs from these samples using horizontal offsets of (0, 1), (0, 2), …, (0, 140). We took the third sample in Figure 7a from a slightly different portion of the plant to obtain a larger number of stipples. Figure 7b shows texture statistics similar to those of Figure 5b. We also performed the same analysis for larger samples of the arrowhead images in Figure 4a, with similar results.

Conclusions

We have shown that stipple distribution statistics vary among hand-drawn, computer-generated, and natural stipple textures. These differences affect the aesthetics associated with the given stipple characteristics. By applying a GLCM texture analysis, we can mathematically define these

Courtesy of Isenberg et al.[9]; copyright respective artists

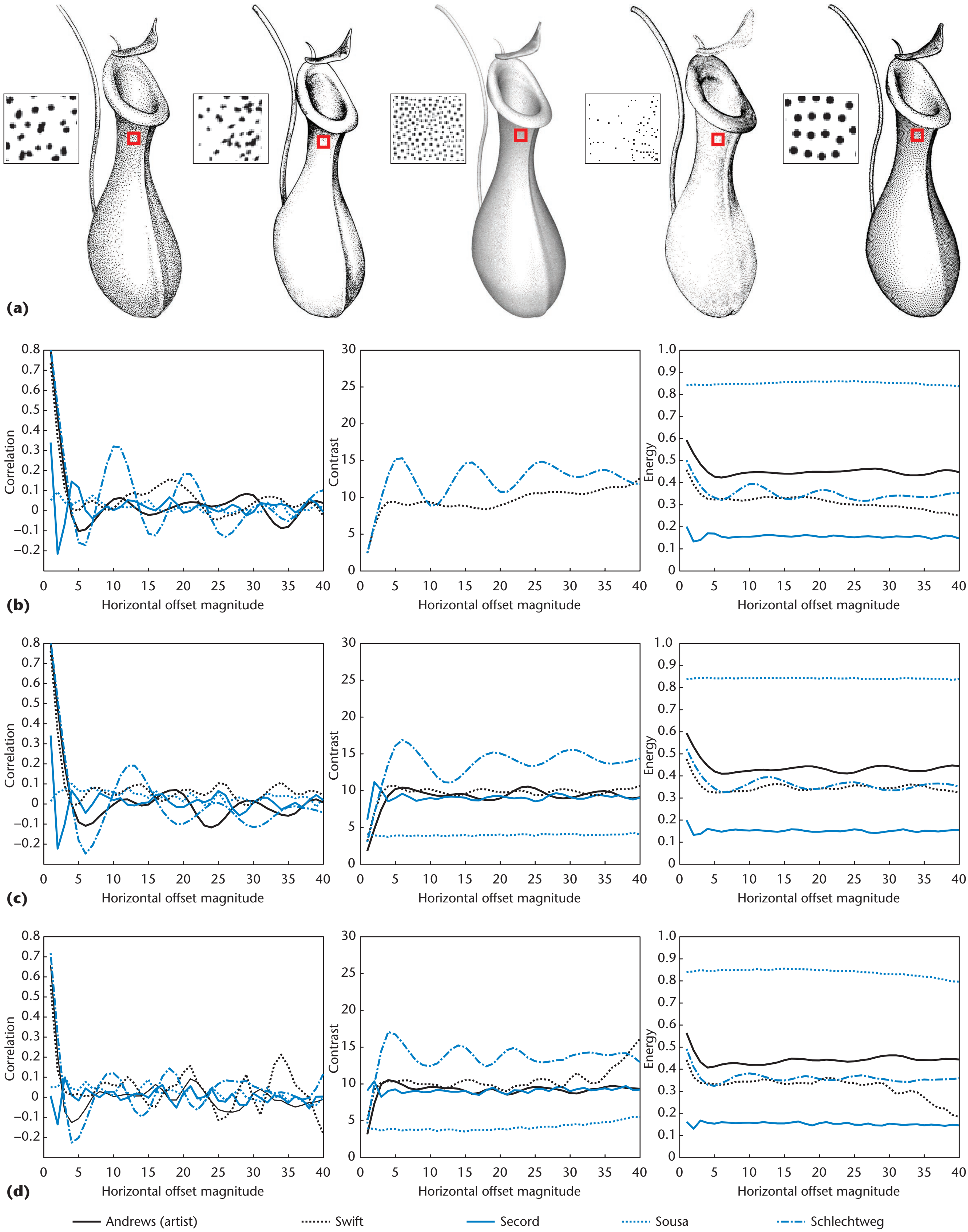


Figure 5. Hand-drawn and computer-generated stippling in botanical illustrations, with varied GLCM offset directions. (a) Stipple images of tropical pitcher plant's trap, with detailed samples. Images from left are hand-drawings by William M. Andrews and Andrew Swift and computer-generated illustrations from Secord, Sousa, and Schlechtweg. (b) Texture statistics with horizontal offset, (c) texture statistics with vertical offset, and (d) texture statistics with southeast offset.

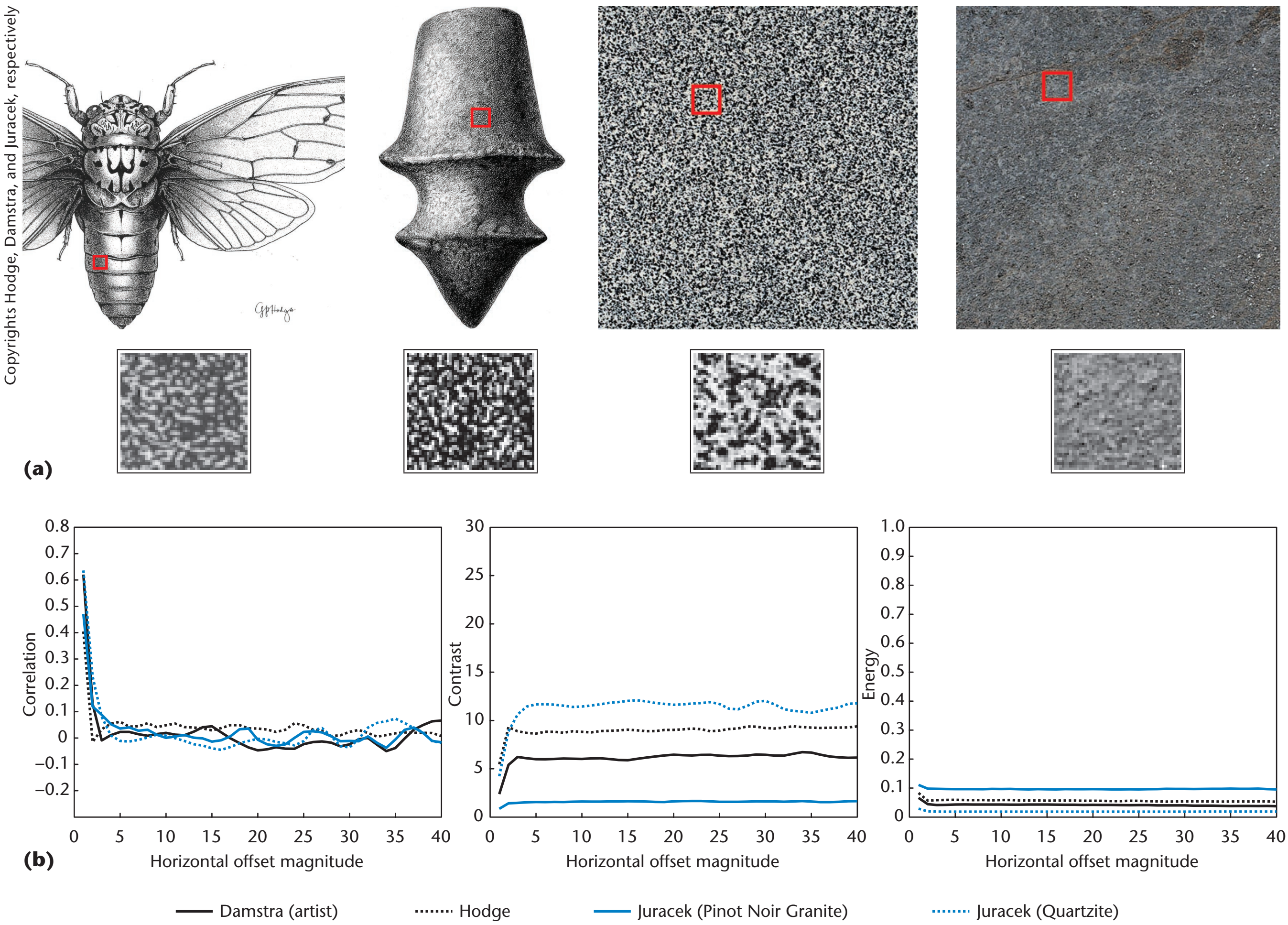


Figure 6. Comparing hand-drawn stippling with natural textures. (a) Hand-drawn and photographic images with detailed samples. Images from left are hand-drawings by Gerald P. Hodge[1] and Emily S. Damstra and photographs of pinot noir granite and quartzite (from Juracek[14]). (b) Texture statistics.

differences and provide quantitative metrics that explain why these different aesthetics occur.

Our results show that hand-drawn stippling has an aesthetic similar to natural textures. Furthermore, with respect to our downsampling normalization, hand-drawn textures favor the placement of stipples within approximately a 5-pixel radius from existing pixels. As artists move out from their seed stipple points, the texture reaches a uniform distribution with a low correlation between the seed stipples and neighboring areas. We found this pattern in all four artists' textures, and although the curves of their textures have slight variations, the underlying pattern remains the same. We also found that natural textures exhibit a strong local correlation among texture pixels, decreasing in correlation as the distance from the source increases.

In contrast, stipple placement in computer-generated textures doesn't follow the same patterns as hand-drawn and natural samples. Computer-generated texture statistics show undesired correlation across the texture or a lack of correlation near the seed stipple point.

To quantitatively describe what we visually observed in the stipple texture correlation statistics, we also measured the correlation coefficient between the GLCM correlation statistics to determine whether there is a linear relationship between our texture samples' correlation statistics. The correlation coefficient ρ of two variables is defined in terms of their covariance and standard deviations as

$$\rho = [cov(X, Y)]/(\sigma_X \sigma_Y)$$

where X and Y are the texture correlation statistics being compared. This gives us a quick and easy way to compare the relationship between two textures' correlation statistics. If no relationship exists, the correlation coefficient is 0. If the correlation statistics match perfectly, the correlation coefficient is 1.

Courtesy of Isenberg et al.[9]; copyright respective artists

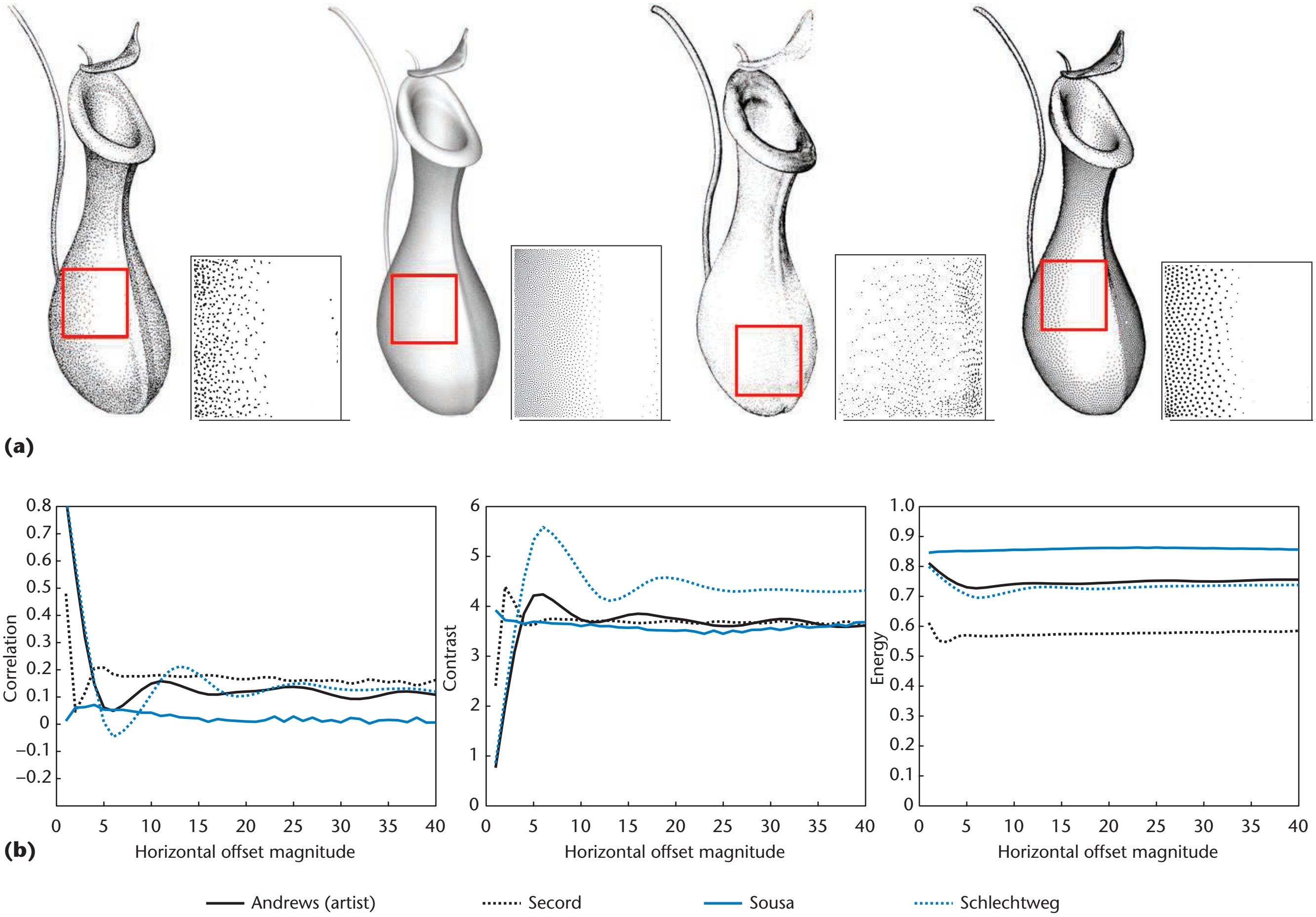


Figure 7. Hand-drawn and computer-generated stippling, with larger sample sizes. (a) Stipple images of a tropical pitcher plant's trap, with detailed samples. Images from left are a hand-drawing by William M. Andrews and computer-generated images from Secord, Sousa, and Schlechtweg. (b) Texture statistics.

Table 1. Correlation coefficients of GLCM statistical correlations from Figure 5b (left).

Artist or technique	Andrews	Swift	Secord	Sousa	Schlechtweg
Andrews	1.0	0.8558	0.2564	0.3031	0.8369
Swift	0.8558	1.0	0.3893	0.3961	0.7459
Secord	0.2564	0.3893	1.0	–0.1064	0.1946
Sousa	0.3031	0.3961	–0.1064	1.0	0.3186
Schlechtweg	0.8369	0.7459	0.1946	0.3186	1.0

We tested the correlation coefficient for all previously discussed plots. Table 1 shows the correlation coefficients for the texture correlation statistics shown in Figure 5b (left). These coefficients are representative of all the other texture correlation statistics discussed in this article. Table 1 shows that the GLCM correlation coefficient between hand-drawn images is large (0.8558), indicating that the signals are closely related. When we compare the hand-drawn GLCM correlation statistics to those of the computer-generated images, we find that the correlation coefficient is very low (less than 0.4), indicating that these signals are not related. However, comparing the hand-drawn GLCM correlation statistics with Schlechtweg's shows that the correlation coefficient is large. You can see this easily in Figure 5b, where both Schlechtweg's and the hand-drawn image plots have smooth curves, falling from a high correlation at an offset of one pixel to a low correlation as the offset approaches five pixels. At the same time, however, it is easy to see that Schlechtweg's texture contains a repeated pattern not present in the hand-drawn stipples.

Overall, current computer-generated stipple textures, generated with mathematical distribution functions such as Poisson and Voronoi, don't compare well statistically with hand-drawn stipples. However, stipple renderers that base stipple placement on factors such as shading and illumination seem to come closer to matching artistic textures. Therefore, generating stipple distribution functions that approximate these local-coherency patterns might match hand-drawn stipple patterns more closely.

Here, we have shown only a small sample of our analyzed stipple textures. The samples we chose for this article are representative of a wide variety of systems. These images provide a realistic comparison since both the artists and the NPR algorithms used the same input—3D models of the skeleton, arrowhead, and tropical pitcher plant. We have other texture samples, as well as samples from different portions of the presented images, that also have the same texture statistics.

Because the GLCM is defined by the orientation **d** while taking into account other statistical texture properties, computer artists could use the GLCM in creating NPR textures. Researchers have already used the GLCM for texture synthesis. Copeland, Ravichandran, and Trivedi presented an analysis of texture synthesis algorithms based on the GLCM of a texture field.[15] They used this work to synthesize pigskin, raffia, and wood textures and conducted a perceptual study to analyze the effectiveness of the synthesis. We plan to extend this work, using our analysis of stippling as the basis of texture generation.

The discrepancies between hand-drawn and computer-generated texture statistics show that there is still room for improvement in computer-generated stipple textures. In the small sample we analyzed, hand-drawn texture statistics seem to have a higher correlation to real textures. From the GLCM statistics we presented in this article, we plan to develop stipple distribution functions and a new set of principles to guide stipple texturing.

Acknowledgments

We thank Emily Damstra, Tobias Germer, Gerald P. Hodge, Aidong Lu, Adrian Secord, and Andrew Swift for providing the stipple images used in this article or the software for producing them. We also thank Edward Delp for his discussions on gray-level co-occurrence matrices. Our work is supported by the National Science Foundation under grants 0081581, 0121288, and 0328984 and by Alberta Ingenuity. It is also supported by discovery grants from the Natural Sciences and Engineering Research Council of Canada. Wei Chen is supported by the National Science Foundation of China under grant 60503056 and by the 863 program of China (2006AA01Z314).

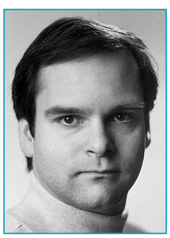
References

1. E.R.S. Hodges, ed., *The Guild Handbook of Scientific Illustration*, Van Nostrand Reinhold, 1989.
2. O. Deussen et al., "Floating Points: A Method for Computing Stipple Drawings," *Computer Graphics Forum*, vol. 19, no. 3, Aug. 2000, pp. 40-51.
3. A. Secord, "Weighted Voronoi Stippling," *Proc. 2nd Int'l Symp. Non-Photorealistic Animation and Rendering* (NPAR 02), ACM Press, 2002, pp. 37-43.
4. B. Wang et al., "Efficient Example-Based Painting and Synthesis of 2D Directional Texture," *IEEE Trans. Visualization and Computer Graphics*, vol. 10, no. 3, May/June 2004, pp. 266-277.
5. R.M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 3, no. 6, Nov. 1973, pp. 610-621.
6. X. Qin and Y.-H. Yang, "Basic Gray Level Aura Matrices: Theory and Its Application to Texture Synthesis," *Proc. 10th IEEE Int'l Conf. Computer Vision* (ICCV 05), IEEE CS Press, 2005, vol. 1, pp. 128-135.
7. S. Schlechtweg, T. Germer, and T. Strothotte, "RenderBots—Multi-Agent Systems for Direct Image Generation," *Computer Graphics Forum*, vol. 24, no. 2, June 2005, pp. 137-148.
8. M. Costa Sousa et al., "Precise Ink Drawing of 3D Models," *Computer Graphics Forum*, vol. 22, no. 3, Sept. 2003, pp. 369-379.
9. T. Isenberg et al., "Non-Photorealistic Rendering in Context: An Observational Study," *Proc. 6th Int'l Symp. Non-Photorealistic Animation and Rendering* (NPAR 06), ACM Press, 2006, pp. 115-126.
10. B. Julesz et al., "Inability of Humans to Discriminate between Visual Textures That Agree in Second-Order Statistics—Revisited," *Perception*, vol. 2, no. 4, 1973, pp. 391-405.
11. J.S. Weszka, C.R. Dyer, and A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 6, no. 4, Apr. 1976, pp. 269-285.
12. D. Chetverikov, "GLDH Based Analysis of Texture Anisotropy and Symmetry: An Experimental Study," *Proc. 12th Int'l Conf. Pattern Recognition* (ICPR 94), IEEE Press, 1994, vol. 1, pp. 444-448.
13. A. Baraldi and F. Parmiggiani, "An Investigation of the Textural Characteristics Associated with Gray Level Cooccurence Matrix Statistical Parameters," *IEEE Trans. Geoscience and Remote Sensing*, vol. 33, no. 2, Mar. 1995, pp. 293-304.
14. J.A. Juracek, *Surfaces: Visual Research for Artists, Architects, and Designers*, W.W. Norton, 1996.
15. A.C. Copeland, G. Ravichandran, and M.M. Trivedi, "Texture Synthesis Using Gray-Level Co-Occurrence Models, Algorithms, Experimental Analysis and Psychophysical Support," *Optical Engineering*, vol. 40, no. 11, Nov. 2001, pp. 2655-2673.

Ross Maciejewski *is a PhD student in electrical and computer engineering at Purdue University. His research interests include nonphotorealistic rendering and visual analytics. Maciejewski has an MS in electrical and computer engineering from Purdue University and BS degrees from the University of Missouri, Columbia. Contact him at rmacieje@purdue.edu.*

Tobias Isenberg *is an assistant professor of computer graphics and interactive systems at the University of Groningen, the Netherlands. His research interests include nonphotorealistic rendering, illustrative rendering and visualization, and the interaction with such graphics. Isenberg did his doctoral studies at the University of Magdeburg, Germany, before completing a post-doctoral fellowship at the University of Calgary, Canada. Contact him at isenberg@cs.rug.nl.*

William M. Andrews *is a certified medical illustrator and an associate professor of medical illustration at the Medical College of Georgia. He is also a PhD student in health promotion, education, and behavior at the University of South Carolina, Columbia. His research interests include health literacy and education, visual perception, and communication theory. Andrews has an MA in biomedical communications from the University of Texas Southwestern Medical Center at Dallas. Contact him at bandrews@mcg.edu.*



David S. Ebert *is a professor and university faculty scholar in the School of Electrical and Computer Engineering at Purdue University, where he is also director of both the Rendering and Perceptualization Lab and the Purdue University Regional Visualization and Analytics Center. His research interests include novel visualization techniques, visual analytics, volume rendering, perceptually based visualization, illustrative visualization, nonphotorealistic rendering, and procedural abstraction of complex, massive data. Ebert has a PhD in computer science from Ohio State University. Contact him at ebertd@purdue.edu.*

Mario Costa Sousa *is an associate professor in the Department of Computer Science at the University of Calgary and a member of the university's Computer Graphics and Visualization Research Lab. His research interests include nonphotorealistic rendering, illustrative visualization, sketch-based interfaces and modeling, volume graphics, and interactive simulations. Sousa has a PhD in computer science from the University of Alberta. Contact him at mario@cpsc.ucalgary.ca.*

Wei Chen *is an associate professor in the State Key CAD and Computer Graphics Lab at Zhejiang University, PR China. He is currently a visiting scholar at Purdue University, where he participated in the work described in this article. His research interests include illustrative visualization, visual analytics, and interactive modeling. Chen has a PhD in computer graphics from Zhejiang university and the Fraunhofer Institute for Graphics, Germany. Contact him at chen23@purdue.edu.*

For further information on this or any other computing topic, please visit our Digital Library at http://www.computer.org/csdl.