https://doi.org/10.2352/ISSN.2470-1173.2020.4.MWSF-023

Signal Rich Art: Improvements and Extensions

Ajith M. Kamath; Digimarc Corporation; Beaverton, OR/USA

Abstract

Signal rich art is an application of watermarking technology to communicate information using visible artistic patterns. In this paper we show new methods of generating signal carrying patterns, simplifications of earlier methods, how to embed a vector watermark signal in applications and how to use signal symmetries to expand the detection envelope of a watermark reader.

Introduction

The goal of digital watermarking technology is to embed covert information in images which can be recovered after naïve image manipulations such as compression, scanning, affine transformation, cropping and noise. The Digimarc watermark signal has been engineered with robust error correction to overcome these common ‘attacks’ on the signal. This makes it possible to express the signal itself in a variety of artistic patterns which enable us to communicate information in plain view, like a UPC barcode or QR code, but with a vastly richer space of available patterns.

In an introductory paper presented last year, we described the idea of signal rich art [3] – of hiding data in artistic patterns in plain view. We showed many examples of signal carrying patterns and described a variety of techniques to generate them in categories such as algorithmic, deep-learning based style-transfer algorithms and image stylization filters.
In this paper, we will present some further advances, including:

- more techniques for generating patterns
- using a vector watermark signal to enhance artwork
- a simpler algorithm for generating patterns like the deep learning style transfer algorithms
- use of signal symmetries to extend the detection envelope of the watermark reader
- Invertible transformations to expand the variety of signal carrying patterns

All the patterns presented in this paper are embedded with the Digimarc barcode, which is a 2D watermark signal designed for use in marking consumer packages to improve the speed of point of sale (POS) checkout, enable consumer engagement and other applications. The ID embedded in the patterns may be read using the free Digimarc Discover application, available for both Apple (iOS) and Android devices. Start the app and hold the phone 3 – 6 inches away from the page and the app should be able to read the ID.
The principles used to generate the signal carrying patterns are very general and apply to any 2D signal which has the following ingredients. We assume the signal is a zero mean binary signal with black and white (BW) pixels representing -1 and 1 respectively. So to have a zero mean, the number of BW pixels must be equal. We also define the signal as a square tile which may be tiled by repetition to cover a larger image or cropped for a smaller image. Figure 1 shows the ingredients: i) a fixed zero mean synchronization signal, ii) a variable zero mean data signal derived from a 1D error correction code with high redundancy on the order of 100 bits (pixels) per information bit. These two signals are combined to produce the signal tile shown on the bottom left and may be seamlessly tiled to cover a larger image as shown by the 3x3 tiling on the bottom right. The number of pixels in the signal tile, its physical size, the sync signal pattern, and the number of information bits all vary in different applications of Digimarc technology.

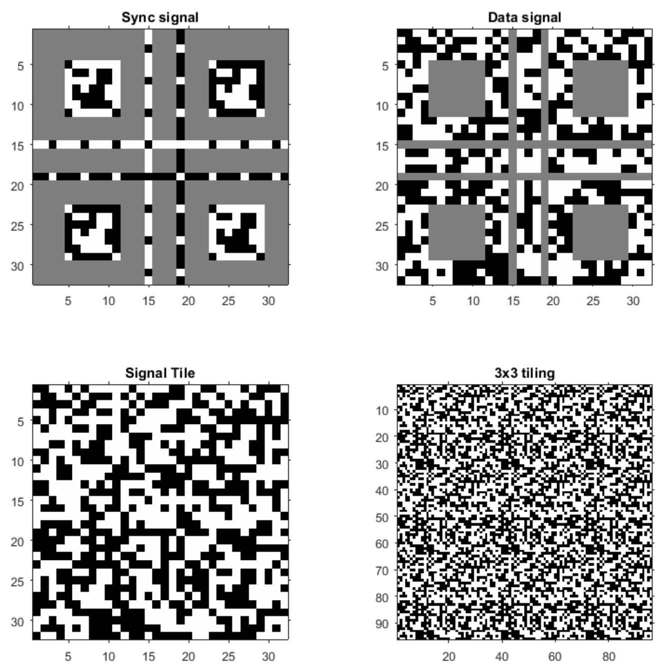

Figure 1 How to design your own 2D watermark signal. Digimarc Barcode uses the same principles in the signal design.

Signal Rich Art

We explain why the idea of signal rich art works in this section. In traditional watermarking, the goal is to embed a data carrying signal in any given cover image. Figure 2 shows the tradeoffs between the number of signal dimensions and the signal energy budget at a high level. Let’s say we have 100 signal dimensions (pixels) to embed our signal. In traditional watermarking we embed the signal at a fixed amplitude in each dimension (this might differ in practice with adaptive/opportunistic embedding). In SRA we embed only (say) 10 signal dimensions, but use 10 times the signal energy in each dimension, since the signal visibility is not a constraint. So the overall energy budget (signal energy per information bit) is the same. However, there are $\binom{n}{k}$ options for choosing a subset of the total, which gives an exponentially large space of signal carrying patterns to choose from. The signal recovery is also helped by the fact that we use an error correction code with a high redundancy (>100x) per information bit, so we can afford to drop 95% of the signal bits and still recover the data reliably from a noisy signal.

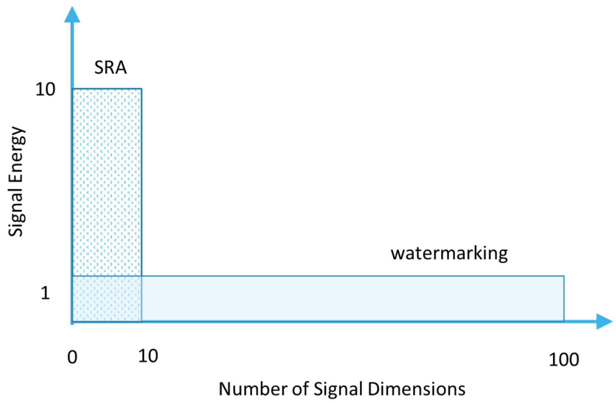

Figure 2 Overview of the tradeoff between signal energy and number of signaling dimensions in watermarking and SRA

Methods and Applications

We present several new techniques for generating patterns, some of which are improvements (faster, more tunable) and extensions (more general, customizable) of previous techniques.

Halftone Patterns

In [3] we showed how to generate signal carrying binary patterns by sparsifying the watermark signal. In this paper we extend this idea to generate halftone dot and line patterns which can be used to generate signal carrying grayscale images. Figure 3 shows a grayscale ramp at the top and two classes of halftone patterns below. The patterns on the left are based on dots of a fixed spatial density (defined by the centers of the dots) and variable dot radii to achieve a grayscale variation. This style is also called amplitude modulation (AM) in the print industry. The line pattern counterparts have a fixed cell size (triangles for Delaunay and polygons/hexagons for Voronoi) and variable line widths. On the right we have dots of a fixed radius, but the density of the dots varies spatially to produce a grayscale variation. This is called frequency modulation (FM) in the print industry. The line pattern counterparts have a variable cell size and fixed line width in this case.

The halftone patterns for the grayscale ramp appear generic, and may indeed be used as a generic halftone pattern tool for any grayscale image. However to carry our signal, we choose the sparse dot locations based on local minima of the grayscale signal, which drives the pattern towards higher correlation with the watermark signal. This increases the complexity of the halftoning algorithm, but allows us to embed the signal along with the pattern as an integrated operation.

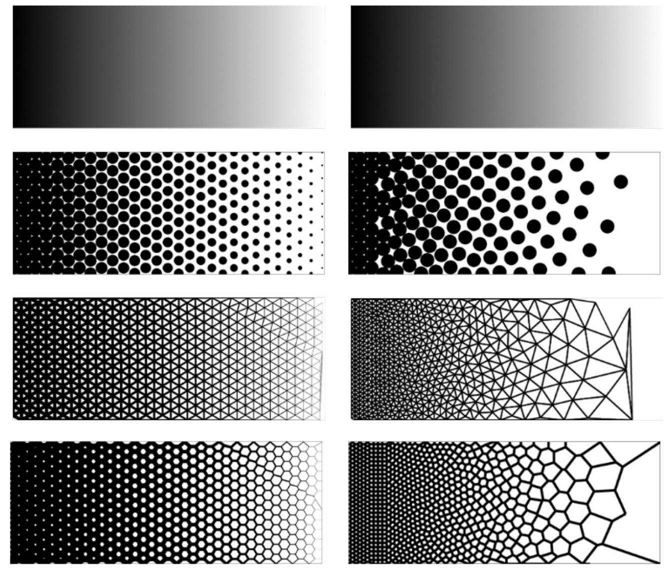

Figure 3 Grayscale ramp created using AM and FM based halftone dot patterns and the corresponding halftone line patterns

Figure 4 shows the halftone algorithm applied to a watermarked grayscale reindeer image with the Voronoi, FM pattern on the left and Delaunay, AM pattern on the right. We had to lower the contrast of the image on the left to achieve a more pleasing cell pattern and also for good signal readability. We are able to maintain the signal strength and contrast with the AM (variable line width) pattern.

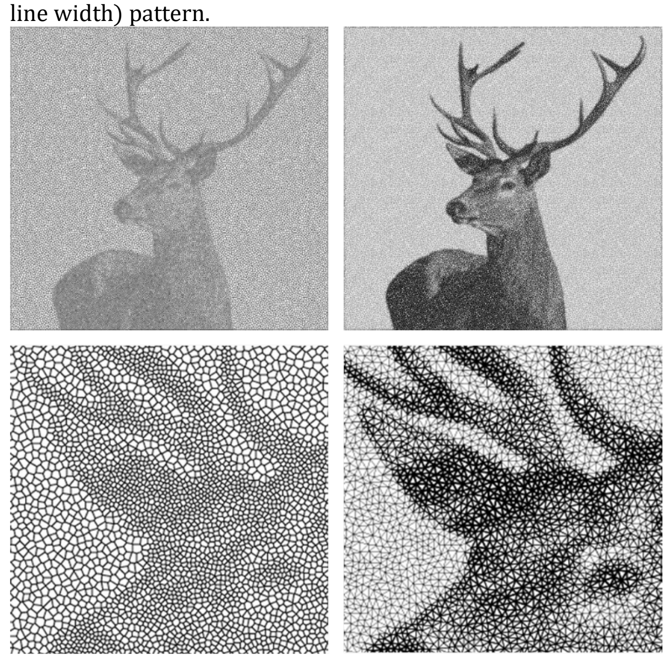

Figure 4 Signal sensitive line halftone patterns applied to a watermarked grayscale Reindeer image

Screening

We may use arbitrary screening patterns to reduce the contrast of a watermarking signal. This also allows us to embed the signal in vector artwork by masking. Figure 5 shows a simple square screen applied to a binary vector signal pattern.

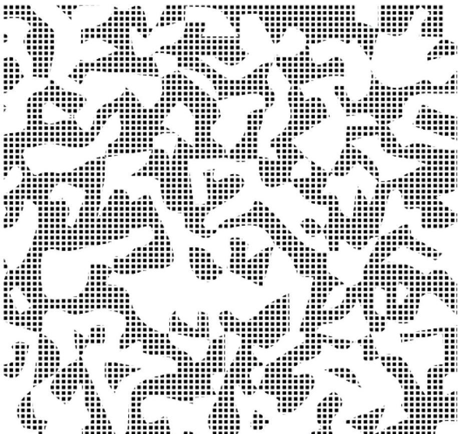

Figure 5 Binary signal pattern masked by a square screening pattern. Arbitrary screening patterns may be used for different artistic applications.

Figure 6 shows a pattern obtained by applying two screening patterns, one consisting of parallel lines applied only to the black pixels in the signal and a second screen with dashed parallel lines applied to the white pixels in the signal. The difference in contrast of the screening patterns is sufficient for the detector to extract the signal.

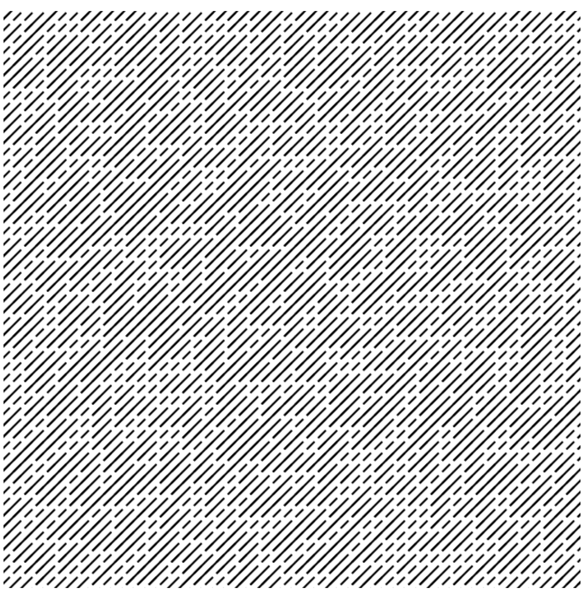

Figure 6 Dual screened pattern

We may further extend this idea to embed two signals in one pattern, by using a signal carrying screening pattern. Figure 7 shows an example where a binary signal pattern is screened using a signal carrying Voronoi pattern which is at 2x the scale. So the information in the screening pattern will be lost at normal detection distance, but it can be recovered at a closer distance. In other words, a low resolution scan will only recover one signal, but a high resolution scan can recover both, by looking for a signal at different scales. This may have applications in forensics and product serialization.

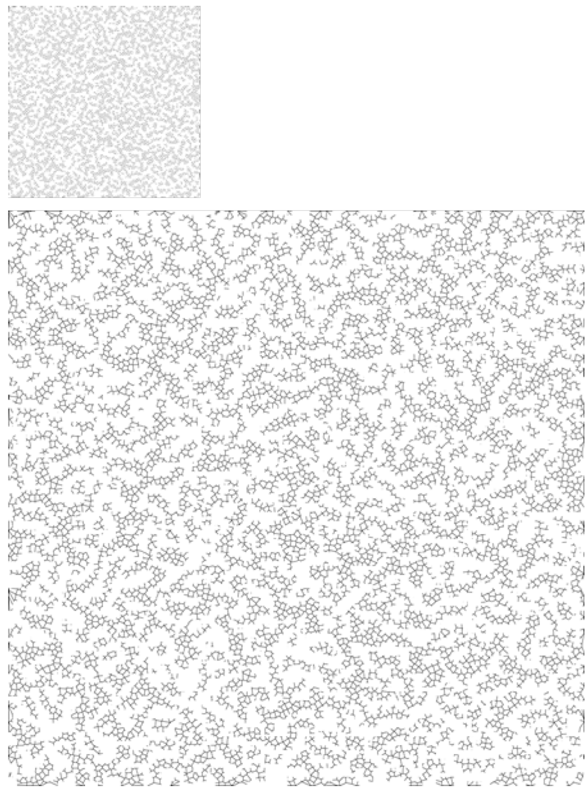

Figure 7 Embedding two signals in one. A signal carrying pattern is used to screen a binary signal tile at 2x the scale. Both the IDs may be recovered by reading the signals at different distances

Vector Enhancement

Figure 8 shows how the screening idea may be applied to create a vector watermark signal which may be used to mark any raster image. This is an exaggerated view to show the details clearly. Given the 'original image' we create two color tweaked images, say one with a magenta tint ('negative') and another with a cyan tint ('positive'). These layers correspond to embedding an all -1's or all +1's watermark signal in the original image, respectively. We then create two screened signal patterns using dots at the locations corresponding to the +/-1 signal bits respectively. The color tweak layers are used to colorize the dots at the corresponding locations and the two layers are merged to form a single 'Signal Mask' layer with 100% opacity at the dot locations and 100% transparency in the gaps between the dots. This layer is overlayed on the original image to obtain the marked image. There are several benefits to this vector signal embedding approach: i) the embedding is DPI independent since the signal is vector and the color tweak raster layers are at the same resolution as the original image, ii) the separation of the signal ID from the embedding allows us to change the ID at any time without manual labor iii) the enhancement layer may be turned on or off to examine the differences.

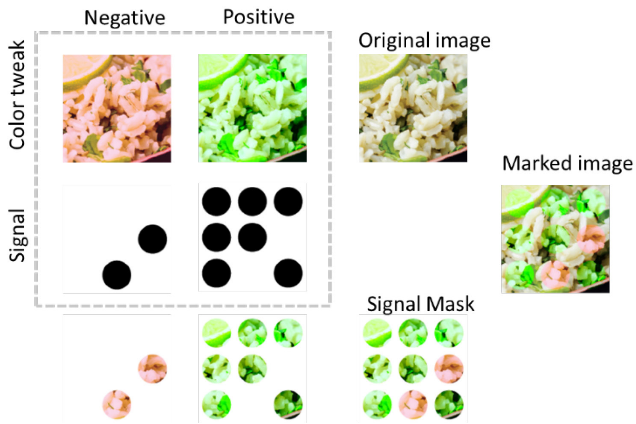

Figure 8 Exaggerated view of a vector dot based watermark signal used to mark a raster image

Figure 9 shows an example of an actual enhancement at a typical signal embedding scale using this method. Here the raster image is 2 inches square when printed.

Figure 9 Example of a package enhanced using a vector watermark layer

Style Transfer SRA from Correlation

Correlating a style image with the signal works almost as well as style transfer for our purposes with these advantages:

- faster, more control over signal level
- does not need special hardware like a GPU. Can run on any CPU

This can also be interpreted as designing 'robust' false positives.

We create a watermark signal carrying image from a given style image by correlating sub-blocks in the signal image with every sub block in the style image as shown in

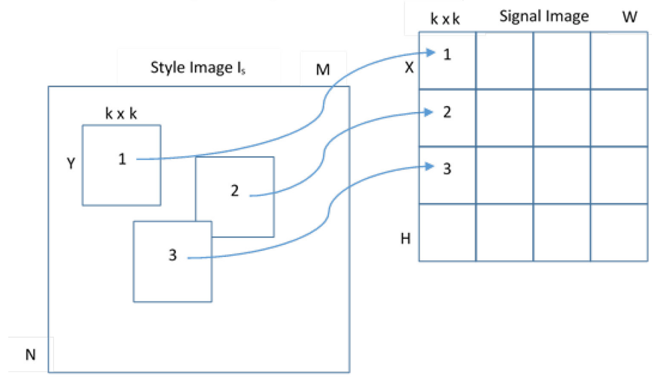

Figure 10 Overview of correlation based style transfer

The style image block with the best correlation is substituted in the signal image to create a signal carrying image, which is also called the mixed image in the style transfer context. This approach is an alternative way of creating a watermarked pattern without modulating pixels in a target image. Since the signal carrying mixed image is a patchwork/collage of the non-watermarked style image, typical watermarking signal visibility criteria need not be considered in this context. The main concern for visibility is the edge discontinuity in the collage.

The size of the signal image block used for correlation depends on the spatial autocorrelation of the style image. If the style image has a very regular pattern, like parallel lines or a repetitive/tiled geometric pattern, then it will have a high spatial autocorrelation and the style image may not be suitable for creating a signal carrying patchwork/mixed image. Other patterns with medium spatial autocorrelation may be used to create a signal carrying mixed image with small blocks, 3% to 6% of the watermark signal tile size (w.r.t. image side length). Patterns with low spatial autocorrelation may be used to create signal carrying images using larger blocks 7% to 15% of the watermark signal tile size. The advantage of using larger block sizes is that the resulting mixed image will have fewer block edge artifacts.

Correlation-Based Style Transfer Algorithm

The cross-correlation between image blocks in the signal and style image is shown in Figure 15 and may be defined as shown below.

$$xcorr(X,Y) = \sum_{j=1}^{k} \sum_{i=1}^{k} (x_{ij} - m_x)(y_{ij} - m_y) \tag{1}$$

For programming convenience and computational speed, it is helpful to rewrite the double summation as a matrix operation. Each column in a matrix B is the vector of k^2 pixels $\boldsymbol{y} = [y_{ij} - m_y]$ in each $k \times k$ subblock of the style image, where m_y is the mean value of the pixels in that block. There are $(M-k)(N-k)$ subblocks of size $k \times k$ in the style image of size MN. Hence, we obtain a matrix of size $k^2 \times (M-k)(N-k)$ by stacking all the vectors $\boldsymbol{y}$ as columns. For large style images, we may prune the number of columns in the matrix B, by choosing a random sub sample of the columns up to a target number of columns.

Similarly, we create a vector $\boldsymbol{x} = [x_{ij} - m_x]$ from the image block X in the signal image.
Then we may write $xcorr(X, Y) = \boldsymbol{x}^T \boldsymbol{y}$, a simple vector inner product. The coordinates u, v of the block in the style image with the best correlation with the current signal image block are given by $\max_{u,v} \boldsymbol{x}^T B$.
The arrows in Figure 11 show the progression of adding blocks to the mixed image.

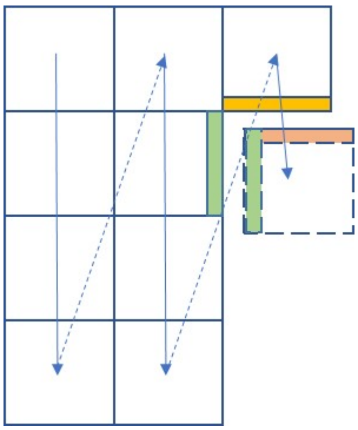

Figure 11 Progression of adding blocks to the mixed image ensuring the edge discontinuities are minimized

To reduce edge discontinuities, we add an additional metric, Total Variation (T.V.), related to the bottom row and right column of pixels of the current mixed image and the top row and left column of pixels of each style image block, respectively, $T.V. = \sum_{i=1}^{k}(|b_i - t_i| + |r_i - l_i|)$. The top and left sides of the last block in the figure are highlighted to indicate the region of desired similarity. To reduce edge artifacts, we want to minimize this T.V. metric, which is the same as maximizing its negative value. Then, we may combine this value with the cross-correlation metric to create a single metric to optimize, using a weighted combination of the correlation and T.V. metrics as follows

$$\max_{u,v}\left[\alpha \boldsymbol{x}^T B - \beta\left(\sum_{i=1}^{k}(|b_i - t_i| + |r_i - l_i|)\right)\right] \quad (2)$$

where the parameters α, β determine the signal strength and embedding smoothness, respectively. Since the result does not change by multiplying both by a common factor, only the ratio of the parameters matters.
For example, we get the most signal, but also the most discontinuous collage, if we choose $\alpha = 1, \beta = 0$. At the other extreme, for $\alpha = 0, \beta = 1$, we get no signal, but a smooth embedding. Figure 12 shows the impact of these choices. To compute the T.V. metric in the left and top row of blocks in the mixed image, we add a dummy left row and top column of gray pixels with a uniform value of 127, midway in the range of pixel values between 0 to 255.

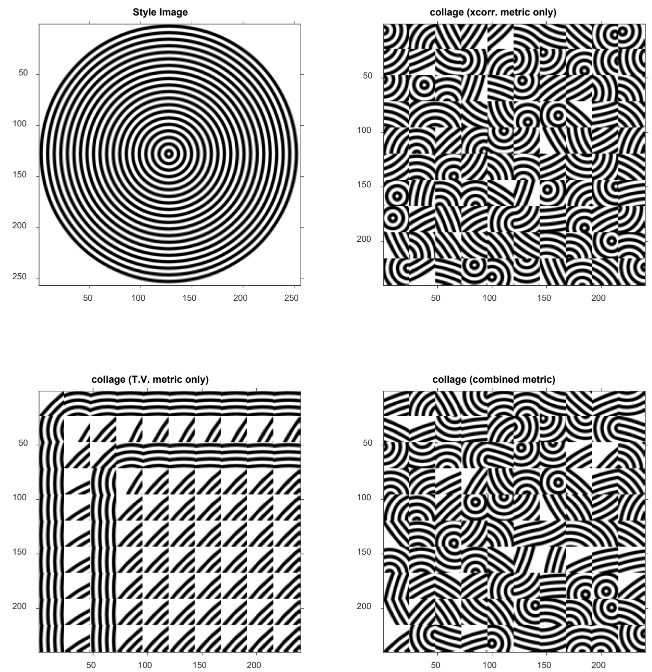

Figure 12 Top right: Using only the xcorr. metric produces a collage with a lot of signal energy, but there are more visible block edge discontinuities. Bottom left: Using only the T.V. metric produces a collage with low edge artifacts, but no signal. Bottom right: the combined metric produces a collage with fewer edge artifacts, but also good signal energy

Figure 13 Correlation based style transfer algorithm applied to a QR code to produce a watermark signal-carrying pattern that mimics a QR code. The QR code synchronization symbols have been added back to complete the pattern. A QR code reader cannot read this pattern, but a Digimarc barcode reader can read it.

Correlation-Based Embedding With Overlap

We extend the idea of correlating a style image with a watermark signal to create a signal carrying collage, to collages with overlapping blocks to reduce edge artifacts.
The main idea is to create a second collage with a half block offset along each axis and create a composite image from a weighted sum of the two collages. This approach is illustrated in the block diagram below, where the green blocks are from the original collage and the orange blocks are from the offset collage.

In practice, we use the toroidal wrapping property of the watermark tile to translate the offset collage so that both the collages are the same size. In Figure 14, the bold orange border shows the second collage and the rest of the orange blocks are obtained by toroidal repetition (i.e., spatial wraparound).

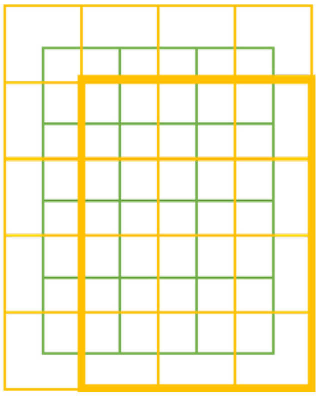

Figure 14. The green blocks are the original blocks. The bold orange border shows the second collage and the rest of the orange blocks are obtained by toroidal repetition (wrap around)

Next, we need to define the weighting function for the blocks in each collage. We need a 2D function such that if it is toroidally shifted by half its side (say L), the sum of the original and shifted functions is a constant. There are many choices for this function, and we choose the following:

$$F(x, y) = \sin^2\left(\frac{\pi x}{L}\right) + \sin^2\left(\frac{\pi y}{L}\right) \tag{3}$$

Then, the function offset by $L/2$ is given by

$$F\left(x + \frac{L}{2}, y + \frac{L}{2}\right) = \sin^2\left(\frac{\pi\left(x+\frac{L}{2}\right)}{L}\right) + \sin^2\left(\frac{\pi\left(y+\frac{L}{2}\right)}{L}\right) = \sin^2\left(\frac{\pi x}{L} + \frac{\pi}{2}\right) + \sin^2\left(\frac{\pi y}{L} + \frac{\pi}{2}\right) = \cos^2\left(\frac{\pi x}{L}\right) + \cos^2\left(\frac{\pi y}{L}\right) \tag{4}$$

Thus from the trigonometric identity $\sin(x)^2 + \cos(x)^2 = 1$, we can show this function has the desired property $F(x, y) + F\left(x + \frac{L}{2}, y + \frac{L}{2}\right) = 2$ for all x and y. Figure 15 shows these complementary weighting functions.

We use this weighting function to weight each subblock in the first and second collage and we add the two weighted collages obtaining the weighted composite. We further divide the sum by 2 because that is the constant value of the weighting function sum. The composite image created this way will have fewer visible edge artifacts compared to the original image.

Figure 16 shows an overview of creating a composite image from a weighted combination of two collage images of a style image. Figure 17 shows the application of correlation-based embedding with edge artifact reduction to an RGB image.

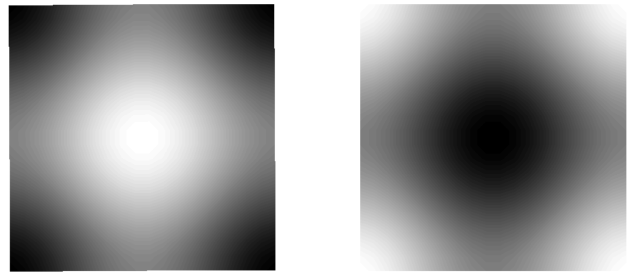

Figure 15 shows the pixel weighting on the left and the complementary weighting function obtained by shifiting the function by half the block width and wrap around.

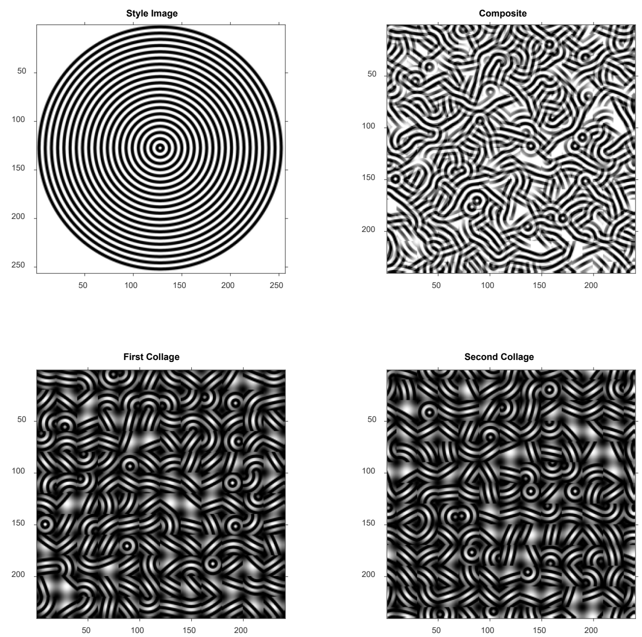

Figure 16 shows the process of creating the composite collage from the first and second collages shown at the bottom with the respective weighting functions applied.

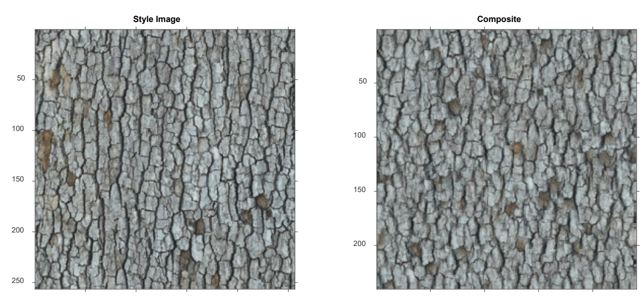

Figure 17 Style transfer-like pattern created using a simple image correlation algorithm. The style image is the tree bark photo on the left and on the right is a signal carrying composite/collage of the same image.

Figure 18 Here we show the correlation style transfer algorithm applied to a heavily watermarked (commercial Digimarc Barcode signal) Lena image shown on the left. The style image is the grass texture shown in the middle. The stylized image on the right does not

DCT and Symmetric Transforms

We may sparsify the signal in a transform domain, such as the DCT to obtain artistic patterns that still carry the signal. A generalization of this idea leads to a richer space of patterns which extend the readability of the watermark signal under mirror reflection and signal inversion (photo negative).

A 2D discrete cosine transform (DCT) is composed of a basis of two-dimensional orthogonal patterns. Figure 19 shows the patterns of basis functions for an 8x8 DCT.

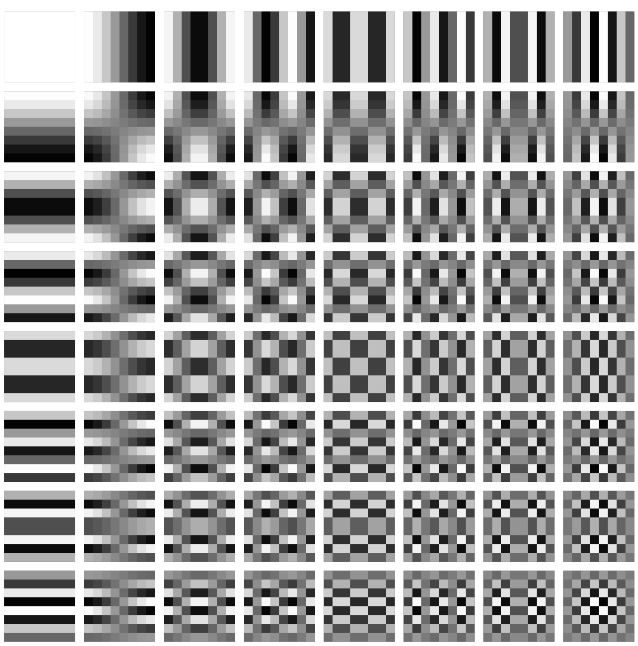

Figure 19 Signal basis patterns of 2D 8x8 DCT

We observe that the patterns with odd indexes ([1,1], [1,3],[1,5],…[3,3].. etc.) are all mirror symmetric along the principal x-axis and y-axis. Hence, if we transform the watermark signal to the DCT domain, then retain only the odd coefficients and apply the inverse transform, we will obtain a pattern with mirror symmetry, because it is formed by the summation of mirror symmetric basis function patterns.

Since we drop a large fraction of the DCT coefficients, this is a sparsification of the signal dimensions in the DCT domain, but there is no additional interference or noise added. If we combine (add) the DCT patterns that map to each other by a transpose of the DCT matrix, the new patterns will have a 90^0 rotational symmetry in addition to mirror symmetry. Then, by projecting the watermark signal to this new basis, we will obtain a sparse representation of the signal with rotational and mirror symmetry.

Figure 20 shows that some of the DCT sparsification patterns may be obtained from purely spatial transformations as well. This approach lets us extend the idea to a richer class of patterns that are also easier to generate. Figure 21 shows some DCT-based patterns that cannot be obtained through spatial transformations.

An additional benefit of the symmetric patterns is they let us expand the range of operation of the detector. For example, we may use the mirror transform to create signal carrying patterns that will detect when the pattern is mirrored. Similarly, using the negative transform lets us create a pattern that will detect if we take the negative of the pattern (an example of this pattern is not shown in this paper).

Some of the other symmetric signal patterns may benefit the detector in a different way by reducing the range of angles over which the detector has to search for the signal. Typically, the detector searches for the signal linear transform over a range of 0 to 180^o, due to the 180^o rotational symmetry of our watermark signal. With the 90^o rotationally symmetric patterns, the detector only has to search over half the angle search space from 0^o to 90^o.

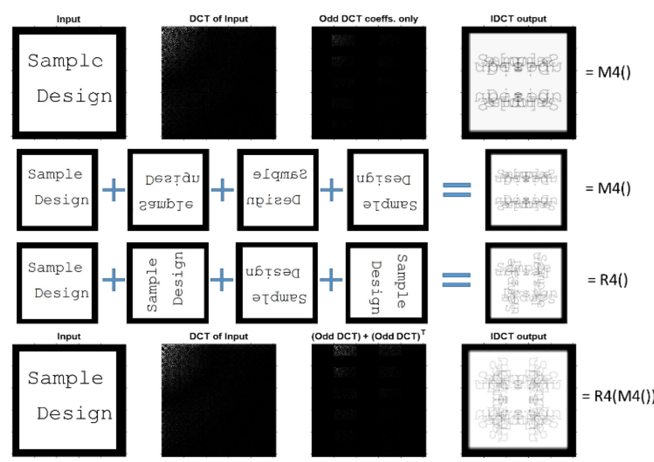

Figure 20 Some examples of DCT patterns that may also be obtained from purely spatial transformations

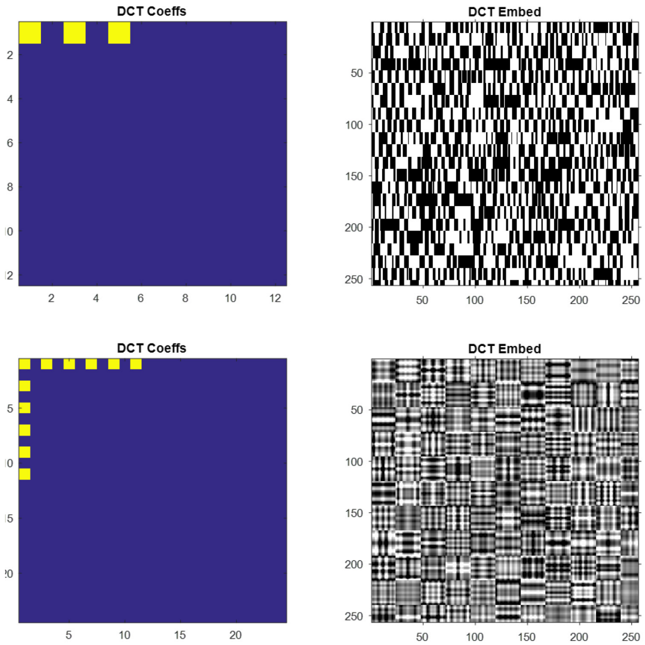

Figure 21 Some patterns that are unique to DCT basis projection

Thus far, we've produced spatially symmetric patterns by adding transformed copies of the signal to itself. We shall refer to these as DCT patterns. An alternative way of producing a spatially symmetric pattern is by using a kaleidoscope. The difference between the DCT and Kaleidoscopic methods is a spatial window function that is applied to the signal prior to applying the symmetry transforms as shown in Figure 22.
There are also differences in the signal robustness of the DCT and Kaleidoscopic patterns. We have observed that at small tiling block sizes (relative to the signal tile size), the DCT window-based patterns are more robust and get progressively weaker as we use larger block sizes. The Kaleidoscopic patterns on the other hand, have a complementary robustness characteristic: larger tiling sub-blocks are more robust than smaller tiling sub-blocks. Furthermore, we may combine (add) the patterns obtained from the DCT and Kaleidoscopic symmetric transforms to obtain not only signal carrying patterns with the same symmetries, but also improved robustness at intermediate sub-block sizes.

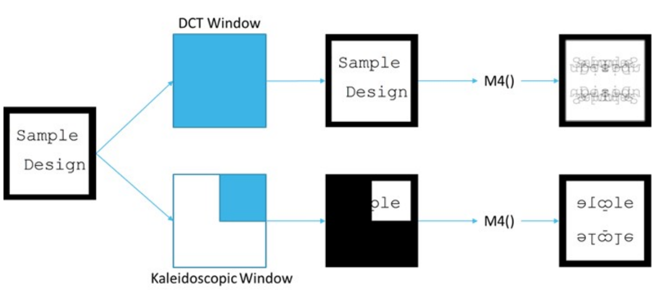

Figure 22 difference between DCT and Kaleidoscopic symmetric patterns

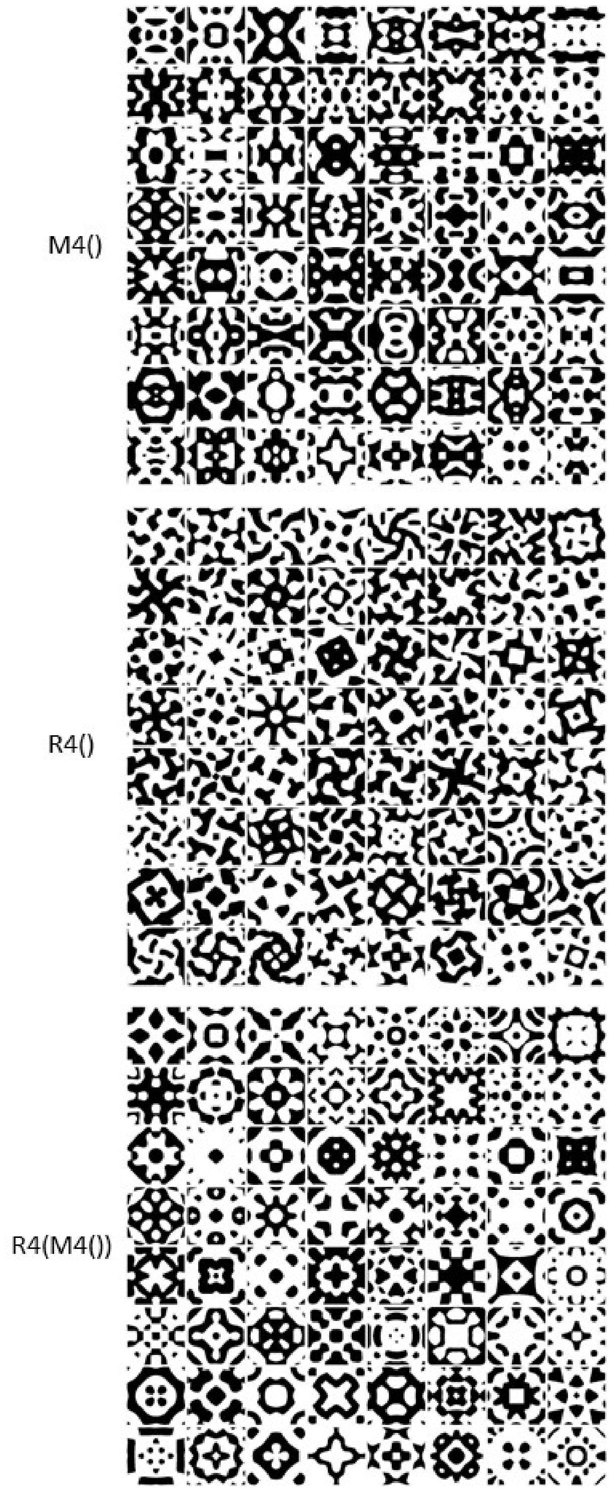

Figure 23 Symmetric patterns obtained by retaining only the odd indexed DCT coefficients and thresholding in the spatial domain.

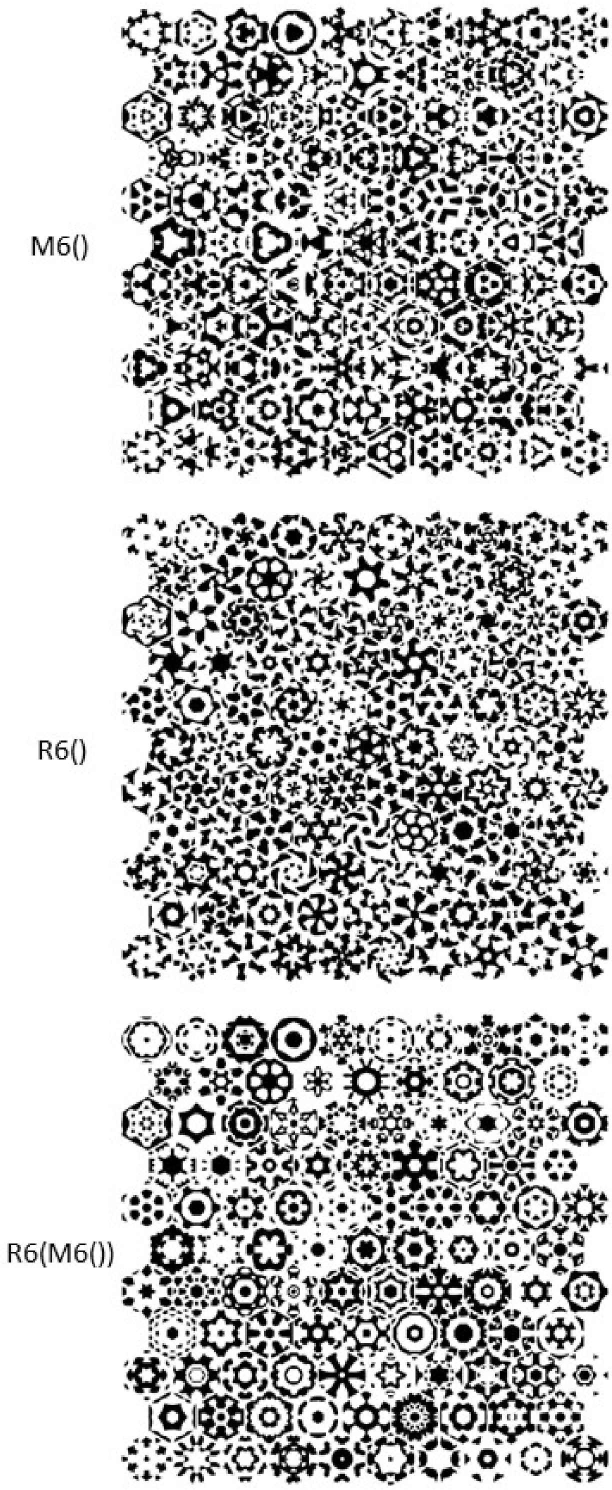

Figure 24 Patterns with hexagonal symmetry, binarized, vectorized (using 'image trace') and denoised using Adobe Illustrator

Invertible Transforms

The idea of sparsifying the signal in a transform domain may be modified to expand the pattern options to a wider variety. We may apply any invertible spatial transform to the signal and create a pattern using any of the techniques we have described before to the transformed signal. We then invert the transform for this pattern in the transformed space, which reverts the signal back to its default scale and rotation. However, the pattern is modified by the inverted spatial transform, thereby yielding a new signal carrying pattern. Figure 25 shows an example in which we create a sparse representation of a watermark signal tile by choosing local maxima (white dots), stretch it 3x horizontally (differential scale transform), compute the Voronoi cells of the dots in the stretched image, and un-stretch the image horizontally (inverse differential scale transform) to restore the signal to its original scale, but this applies the inverse transform to the Voronoi cells thereby producing a new signal carrying pattern.

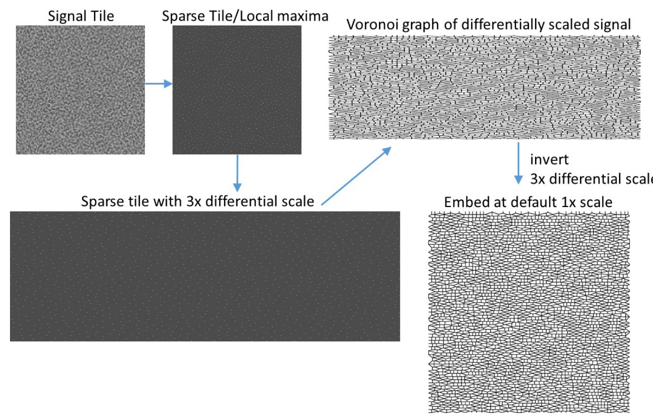

Figure 25 Differential scale & Voronoi pattern

Figure 27 shows another example in which we apply an invertible conformal transform combined with the hexagonal symmetric tiling pattern.

Figure 26 shows an example in which we use a TSP path to connect the dots in a sparse watermark signal (local minima) to obtain a signal carrying pattern [4]. We don't need the TSP solver to find an optimal path in this case, any non-self-intersecting path through the points will do. We may create the TSP path using a piecewise linear path as shown at the top right, or interpolate the path between the points using splines or an FFT based interpolator [7]. If we duplicate each point and apply the FFT interpolation, we will obtain small loops near each point on the path, which improves the signal correlation.

We may further use the invertible transform idea to generate two TSP paths based on differentially scaling the sparse points by 3x vertically and horizontally and obtain a pair of TSP paths which intersect at the locations of the sparse points. Using the FFT interpolation with repetition, we obtain the pattern on the bottom left and without repetition, the pattern on the bottom right.

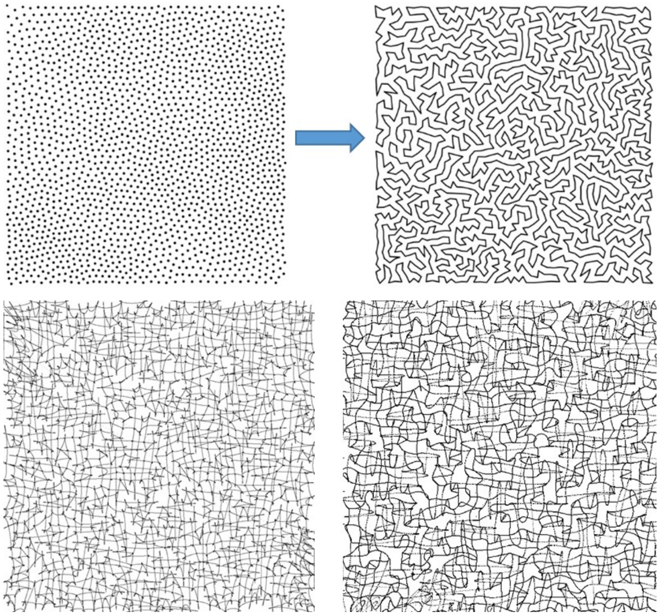

Figure 26 TSP pattern from sparse dot pattern

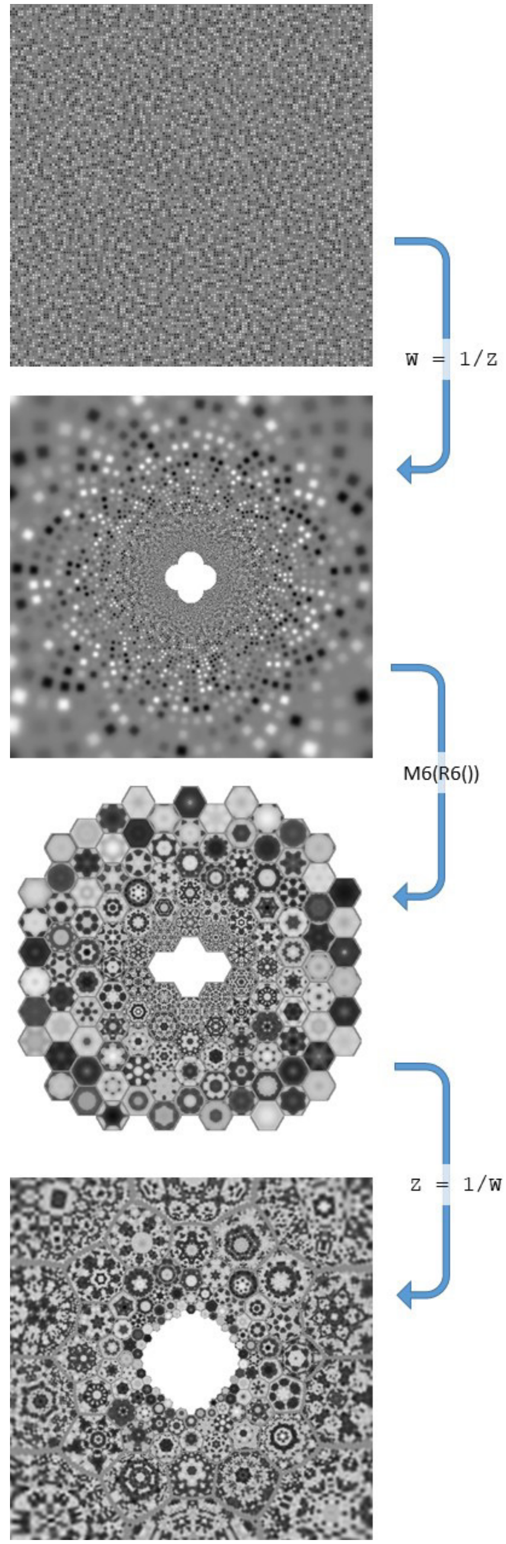

Figure 27 conformal invertible transform

Conclusion

We introduced the concept of signal rich art in [3]. In this paper we have described some more techniques of generating signal carrying patterns and new applications of the concept. We described how signal carrying patterns may be generated from 2D signals with high error correction code redundancy, by choosing a subset of signal dimensions to put the energy in. Since the visibility of the signal carrying patterns is not a concern, we could potentially embed a watermark signal more strongly than in a typical covert embedding application.

In particular, the watermark screening idea may be applied to create a vector watermark signal which may benefit the consumer package enhancement workflow.

The techniques described here - halftoning, screening, DCT, invertible transforms - may all be combined in different ways to obtain a larger variety of signal carrying patterns.

References

[1] B. Davis, "Signal rich art: Enabling the vision of ubiquitous computing", *Proc. SPIE*, vol. 7880, pp. 788002, Feb. 2011.

[2] Alattar, Adnan; "Smart Images" Using Digimarc's Watermarking Technology; IS&T/SPIE's 12th International Symposium on Electronic Imaging, San Jose, CA, Jan. 25, 2000, vol. 3971, No. 25, 1.

[3] Kamath, Ajith M.; Palani, Harish, "Hiding in Plain Sight: Enabling the Vision of Signal Rich Art", Electronic Imaging, Media Watermarking, Security, and Forensics 2019, pp. 527-1-527-8(8)

[4] Craig S. Kaplan and Robert Bosch. “TSP Art.” In Renaissance Banff: Bridges 2005: Mathematical Connections in Art, Music and Science, pages 301-308, 2005.

[5] Bosch, Robert. “Opt Art: From Mathematical Optimization to Visual Design,” Princeton University Press, 2019.

[6] https://github.com/donpark/identicon

[7] Sanderson, Grant. "But what is a Fourier series?" 3Blue1Brown series S4.E4. https://youtu.be/r6sGWTCMz2k

Author Biography

Ajith Kamath received his BTech in Electrical Engineering from IIT Madras (2000) and PhD in Electrical Engineering from North Carolina State University(2005). He is currently working as an R&D Engineer at Digimarc Corporation since 2009. His work focuses on the development of embedding and detection algorithms, as well as novel applications of watermarking.