

Application of Self-Organizing Maps to Texture Visualization and Analysis¹

By: Jacob Ratkiewicz

Abstract

The analysis of visual texture occupies a central role in many areas of computer vision. Applications of this technology include product inspection, medical image analysis, and analysis of remotely sensed data, such as satellite images of the Earth. Standard approaches to texture analysis generate a large number of statistics which are difficult to intuitively visualize. We explore the application of Self-Organizing Maps (SOM), a type of artificial neural network, to visualizing these texture descriptors.

1. Introduction

Visual texture, while an intuitive concept to grasp, is one that is difficult to rigorously define and quantify. Texture has been characterized as being made up of "repetitive patterns," [1] or possessing a "constant, slowly varying, or approximately periodic" set of local statistics. [2] Some examples of texture images are shown in Figure 1. Various techniques for analyzing texture data have been proposed and used with some success. These include geometrical, model-based, signal processing, and statistical methods.

1.1 Geometrical Methods

Geometrical methods center mainly on their characterization of textures as being constructed of "texture elements" in the form of primitive geometrical shapes. Analysis of a texture is then a problem of extracting these texture elements and either characterizing them individually in terms of statistics derived from them, or describing the original texture in terms of the placement of these texture elements within it. [3]

1.2 Model-Based Methods

To analyze a texture using a model-based method, a theoretical model of the texture's perceived structure is generated, which ideally describes the texture itself as well having the ability to synthesize new textures of the same structure. Markov random fields are an example of a useful model based approach [3].

1.3 Signal-Processing Methods

Signal processing methods, such as those utilizing wavelet techniques, have also been successfully utilized for texture analysis. Texture apparently lends itself well to signal processing analysis. Recent studies also suggest that the human brain may perform frequency analysis on the images it perceives [3].

1.4 Statistical Methods

Statistical methods have also proven useful. Sharma *et al.* [11] evaluate a number of texture analysis methods, and conclude that statistics derived from co-occurrence matrices yield the best results of any single method, when tested against the auto-correlation, edge frequency, primitive-length, and Law's-method techniques. A more detailed description of the construction of these matrices and the statistics based upon them follows.



Figure 1 - Some examples of visual texture. Note the regular patterns, uniform brightness, and geometric forms present. While intuitively we can class all of these as visual texture, it's hard to precisely define the characteristics that they have in common.

2. Gray Level Co-Occurrence Matrices

For the $N \times N$ image with G gray levels and a displacement vector d , Haralick[4] defines the *gray level co-occurrence matrix* P_d as the matrix in which the (i, j) entry is the number of times that the gray value i follows the gray value j with displacement d . For example, in the artificial GLCM in Figure 2b, the number 2 in the (1,2) position represents the fact that the gray value 1 followed the gray value 2 with displacement d twice on the sample image in Figure 2a, indicated by the diagonal lines.

Artificial image subset			
1	2	1	3
2	2	1	1
1	2	3	2
1	3	3	1
$d.$			

Resulting gray level co-occurrence matrix

	1	2	3
1	0	2	1
2	2	1	1
3	1	0	1

Displacement $d = \langle 1, 1 \rangle$

$b.$

Figure 2 - An artificial image and its corresponding GLCM

The GLCM consolidates information about the distribution of gray levels in an image, allowing analysis of the data for patterns it may contain. As a part of his analysis, Haralick also proposed a set of useful statistics derived from the co-occurrence matrix. Among those were the following, also used in our experiments: Angular Second Moment, Contrast, Correlation, Inverse Difference Moment, Entropy, Variance, Cluster Prominence, Cluster Shade, and Diagonal Moment. These statistics codify in numerical form various

visual characteristics of the image; for example, Homogeneity is the numerical measure of the image's tendency to have smooth changes in color.

These statistics capture a wide range of the image's features. In practice, a set containing these nine statistics is computed for each image, and is represented as a nine-dimensional vector. Table 2, for example, shows the statistics generated for the images shown in Figure 1.

Visualizing these vectors, or image feature descriptors, was the primary goal of this project. Specifically, we explore the applications of Kohonen's self-organizing feature map as a visualization tool for these descriptors.

Angular Second Moment	$\sum P_d^2(i, j)$
Contrast	$\sum_{i,j} (i-j)^2 P_d(i, j)$
Correlation	$\sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)}{\sqrt{\sigma_i^2 \cdot \sigma_j^2}}$
Inverse Difference Moment	$\sum_{i,j \neq j} \frac{P_d(i, j)}{ i-j ^k}$
Entropy	$\sum_{i,j} P_d(i, j) \log P_d(i, j)$
Mean	$\mu_i = \sum_{i,j} i \cdot P_d(i, j)$, $\mu_j = \sum_{i,j} j \cdot P_d(i, j)$ $\mu_{ij} = \mu_i + \mu_j$
Standard Deviation	$\sigma_i^2 = \sum_{i,j} P_d(i, j) [i - \mu_i]^2$ $\sigma_j^2 = \sum_{i,j} P_d(i, j) [j - \mu_j]^2$
Variance	$\sigma_i = \sqrt{\sigma_i^2}$, $\sigma_j = \sqrt{\sigma_j^2}$
Cluster Prominence	$\sum_{i,j} P_d(i, j) (i + j - 2 - \mu_{ij})^4$
Cluster Shade	$\sum_{i,j} P_d(i, j) (i + j - 2 - \mu_{ij})^3$
Diagonal Moment	$\sum i-j \cdot (i + j - 2 - \mu_{ij}) P_d(i, j)$

Table 1 - Formulae for GLCM Statistics

3. Self-Organizing Feature Maps:

The concept of self-organizing maps of neural networks was first developed by Teuvo Kohonen in the early 1980's [5-8]. It has found application in fields as diverse as the analysis of recorded human speech, the processing of images of the Earth taken from space, and the analysis of medical data. The basic SOM can be thought of as a sheet-like neural structure, in which each neuron contains a vector of the same dimensionality as the input data. The map is first initialized with random vectors, then "trained" by the following iterative process – Figure 3 shows an example.

- a) An input vector, x , is presented to the map.
- b) The map is searched for the component neuron containing the vector most similar to x , as determined by a metric such as the Euclidian distance function.
- c) The closest vector, Mx , is located and modified to be incrementally closer to the vector x .
- d) A neighborhood Nd about Mx is then defined as being all those neurons that are within distance d of Mx , and each neuron in that neighborhood is caused to become slightly more like x as the neighborhood radius d is decreased iteratively. Thus, those neurons close to Mx become more like x , but do so to a greater degree than those farther away.

Mathematically, the process is carried out as follows. Let $t = 1, 2, \dots$ be the step index, and determine each best matching index c for each sample $x(t)$ by the following condition:

$$\forall i, \|x(t) - m_c(t)\| \leq \|x(t) - m_i(t)\|$$

Following determination of this "winning" node $m_c(t)$, a subset of the nodes around it determined by the "neighborhood function" $h_{c(x),i}$ are updated according to the following:

$$m_i(t+1) = m_i(t) + h_{c(x),i}(x(t) - m_i(t))$$

This regression is typically performed iteratively over available samples until a satisfactory map is formed – that is, one that separates the vectors that are used to train it into some kind of grouping that the viewer can make sense of.

Since the initial vector placement on the map is random, it is typically necessary to generate a large number of trial maps before obtaining a map suitable for visualization. Kohonen *et al.* developed procedures to automate this map selection process. Upon completion of this algorithm, we expect to find similar vectors, representing similar features, clustered close together in the final map. In general, we may view the SOM as a way of mapping a higher dimensional space to a lower dimension, while preserving much of the order inherent in the higher-dimensional structure.

4. Computing Environment

For our experiments we used standard AMD Athlon 1.2 GHz PCs with 256 MB of RAM running the RedHat distribution of GNU/Linux. For image preprocessing we used the Xite [12] suite of tools as well as custom scripts written in Perl and Python. To generate maps we used the SOM_PAK [13]. A typical high-quality map took an average of four to six hours to generate with this hardware/software combination.

5. Visualization of Baseline Textures

For our initial experiment we chose textures available as part of the VisTex visual texture library [9]. We partitioned these textures into a training set and a testing set, selecting sample textures of each visual description (i.e. clouds, water, tile, etc) for each set. We then performed the following process on each set:

1. Crop images to a uniform size.
2. Convert images to a grayscale representation in the proper file format.

ASM	Contrast	Correlation	IDM	Entropy	Variance	Diag. Moment	Cluster Shade	Cluster Prom
0.1062708	0.03366648	1.478081e-05	0.5231162	0.7864788	0.2731719	0.2929115	0.2648529	0.1409712
0.01182424	0.06037223	4.531076e-05	0.2036668	0.8939893	0.5660633	-0.04692755	-0.8037589	0.5553433
0.02734371	0.03249223	2.972172e-04	0.4728295	0.8215885	0.2038240	-0.03416768	-0.1012733	0.07205668
0.02264220	0.04929722	0.001291626	0.2284742	0.8310419	0.08228722	0.1562082	0.03632461	0.01390445

Table 2 - Sample vectors from the images shown in Figure 1

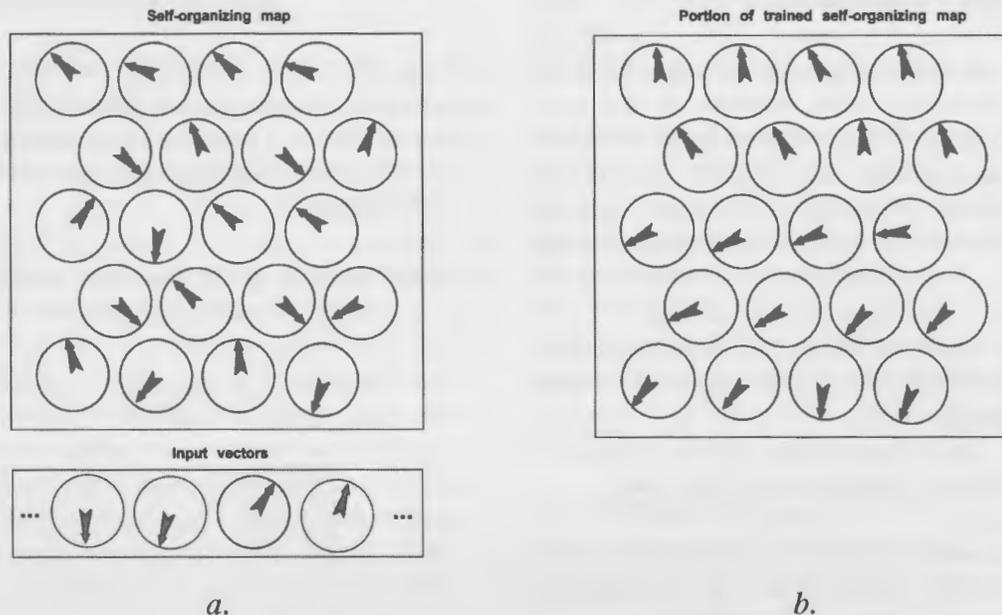


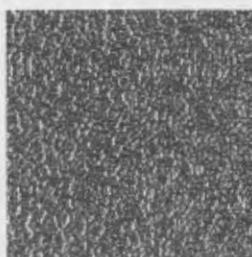
Figure 3 - An artificial self-organizing map, before and after training

3. Generate the nine GLCM statistics for each image, and place the resulting vectors in a data file – each line of the data file containing the nine vector components representing one texture.
4. Normalize each column vector in the data file.
5. Train the self-organizing map on the data interactively.²
6. Display the map using a umatrix visualization. In a umatrix the map is represented as a grid of nodes. Each node is color-coded to represent the average distance between it and its neighbors, and labeled with the label of the input vector that it most closely matches (if there is one) or with a dot (if it matches no input vector).

The application of these steps is illustrated in the following simple example. First we selected three textures from the VisTex database that were reasonably distinct from each other, as shown in Figure 4. Since three images represented by three vectors provide insufficient data for effective visualization, we subdivided each texture into 16 smaller texture images, each having statistics similar to those of the parent image. We used the resulting vectors to train a self-organizing map consisting of 300 nodes arranged in a 30x10 hexagonal grid. Figure 5 shows the resulting u-matrix visualization. Note the clustering of the textures by visual similarity.



Texture A (*Fabric*)



Texture B (*Metal*)



Texture C (*Wood grain*)

Figure 4 - Images used in this experiment

ASM	Contrast	Correlation	IDM	Entropy	Variance	Diag. Moment	Cluster Shade	Cluster Prom
0.9684827	0.005723557	1.000000	0.9919758	0.6103299	.006407373	-0.0004233161	-8.991040e-05	3.377443e-05
0.03910838	0.8241672	2.457828e-05	0.1977914	0.9567972	0.8525648	0.8815748	0.7389579	0.6972131
0.03963804	0.8263306	2.675579e-05	0.1964169	0.9556544	0.8058157	0.8879519	0.6914841	0.6412812
0.02250192	0.6140333	9.628698e-05	0.1025383	0.9899384	0.4590231	0.1435912	0.06562232	0.1324560
0.7320469	.008291456	0.6378333	0.8320489	0.6406141	0.007660096	0.001709761	7.052707e-05	4.966373e-05
0.6815332	0.01102695	0.4283442	0.7531977	0.6507503	0.009145611	0.003249236	5.739477e-04	1.979026e-04
0.02018491	0.8281551	6.869980e-05	0.08987050	1.000000	0.5945056	0.1409244	0.06873654	0.2037111
0.02281073	0.6249805	9.906084e-05	0.1104830	0.9886962	0.4507733	0.1387794	0.06166328	0.1239210
0.02348824	0.5844237	9.939955e-05	0.1123659	0.9886962	0.4071553	0.1689786	0.06610873	0.1022450

Table 3 – Nine of the 48 statistic vectors produced.

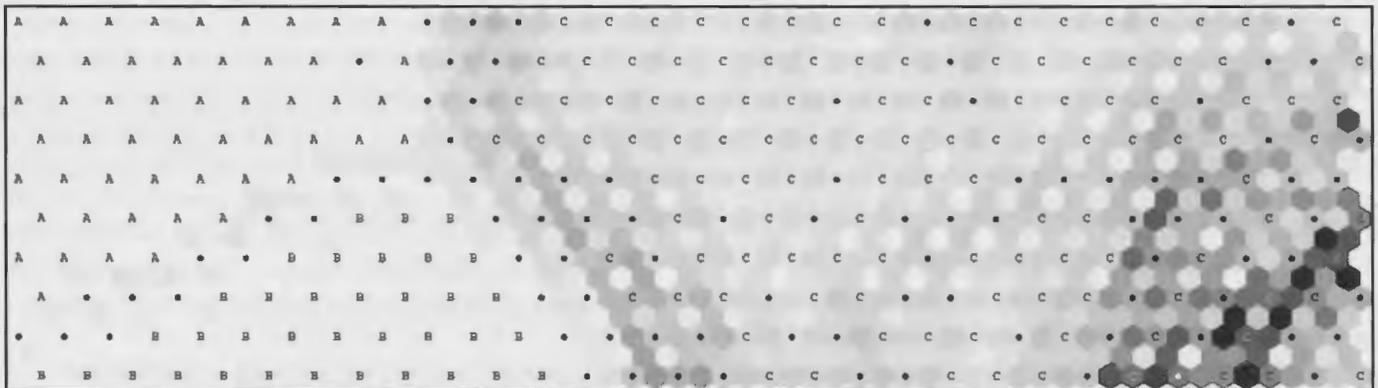


Figure 5 - Scaled statistic vectors from A, B, and C, including those in Table 3, visualized using a self-organizing map. Note the clear demarcation between vectors from the sub-images of the three textures.

Following these initial experiments we expanded our study to include the testing and training data sets described at the beginning of section 5. Our goal was to see if the SOM segregated the statistical vectors in a manner that reflected the original image represented by the vector – i.e. do vectors representing similar visual images cluster together in the resulting display. To answer this question, we performed the following experiment:

1. Generate three self-organizing maps of differing dimensions and parameters, but all with the training set as their input data.

2. Map the testing set onto each of these maps.

3. Evaluate the results by the following criterion: If the texture in the training set to which the training vector mapped the closest appeared visually similar³, count the match as a success; otherwise, count it as a failure.

In these experiments, the SOM typically mapped previously unseen textures between 70% and 80% of the time. This is consistent with research literature, reporting the discrimination capability of the GLCM-derived statistics at about 80% on similar data sets [11].

To assess the scalability of this approach, we subdivided the input textures into smaller sub-images (thus scaling up the number of vectors that the map would need to use as its training set). We found that the SOM algorithm was unable to discern the minute differences in the several thousand very similar input vectors, even when the number of nodes was also increased. Figures 10 and 11 show two SOMs. The first is trained on a vector file that represents each texture in the training data set split into four sub-images; the second represents each texture in the training data set split up into sixteen sub-images. Note the strong contrast between the two maps in terms of their ability to group the data coherently. Figure 11 shows the image statistic vectors separated into clusters on the map, grouped roughly by visual similarity. In Figure 10, by contrast, most of the vectors are packed tightly together with only a few deep ravines running across the lower left-hand corner of the map. We believe this is due to the insensitivity of the GLCM in conjunction with SOM visualization to very small degrees of separation in the input vectors.

6. Visualizing Fingerprint Texture Statistics

Having explored the effectiveness of the self-organizing map algorithm for visualization of statistics derived gray level co-occurrence matrices on a baseline set of textures, we sought to explore its possible use in visualizing the texture statistics of human fingerprints. The fingerprint images we used for this experiment were acquired from the publicly available fingerprint image sample set on from the National Institute for Standards and Technology (NIST) [10]. As illustrated in Figure 6, we discovered that many of the fingerprints collected are direct scans of fingerprint cards, complete with handwritten notes, labels (e.g. "R. Thumb") and are of inconsistent size and orientation. Additional preprocessing was necessary during the preparation of these images to remove handwriting (where possible) and to form images of consistent size and orientation.

Also, since these images were acquired under various lighting conditions, we performed histogram normaliza-

tion on each image to standardize brightness and contrast. This involved modifying the histogram of each image through a Gaussian normalization process. After normalization was complete, we continued with the generation of the GLCM statistics.

We approached this problem in two stages. First we tested the SOM algorithm's ability to discriminate the difference between vectors representing fingerprint textures and those representing the textures from the VisTex database. Figure 8 shows one of the resulting maps, with fingerprint textures labeled "F" and all others labeled "T". Note that the vectors representing the fingerprint images are reasonably well separated from the other textures, evidence that the vectors representing their GLCM statistics are sufficiently distinct.

Further experiments, however, seemed to indicate that individual fingerprint textures as representing by GLCM statistic vectors are not sufficiently distinct from one another to allow meaningful classification with this technology. Fingerprints are typically classed by features such as the shape of the center structure, often described as a "loop," "whorl," or another such category. The conjunction of gray-level co-occurrence matrices with the self-organizing map does not seem to be sensitive to structural elements at this level of detail, and therefore we do not observe fingerprints with the same geometric characteristics (i.e. loops or whorls) close to each other with any significant frequency. This is consistent with the experiments described in section 5, in that the SOM is unable to distinguish reliably between textures which are only subtly different. Figure 9 shows one of the maps produced on the set of fingerprint textures.

7. Visualizing Merged Fingerprints and Textures

Having established the capability of the self-organizing map for visualizing GLCM statistics derived from individual textures, we explored visualizing statistics obtained from images composed of fingerprints and various other textures. Our goal was to determine if statistics representing the composite textures were sufficiently distinct



Figure 6 - Fingerprints from the NIST database, prior to preprocessing

from statistics from the other two types for visualization to be possible with a SOM.

The textures we used in this experiment were generated by overlaying each of forty randomly-chosen fingerprint textures on forty of the standard VisTex textures, one-to-one, to produce forty composite textures. This blending was accomplished by choosing the darkest of the pixels in each of the source images to produce the destination pixel; we chose this method to enforce transparency of the light areas of the fingerprint images, and give the visual appearance of a fingerprint impressed on a surface with some dark ink. Figure 10 shows some of the

resulting textures. We realize that a fingerprint such as the one in Figure 10 (A) – e.g. a fingerprint on bark at this relative scale – is unrealistic. Fingerprints in the real world are most often detected on smooth surfaces. We chose fingerprints such as those in Figure 10 because of the interesting technical challenges they presented rather than from the possibility of any real-life application.

Following the construction of these images we generated their representative statistics as described previously, then trained a self-organizing map using these statistics in conjunction with those from the other two texture



Figure 7 - The fingerprints from Figure 6, after preprocessing

<i>ASM</i>	<i>Contrast</i>	<i>Correlation</i>	<i>IDM</i>	<i>Entropy</i>	<i>Variance</i>	<i>Diag. Moment</i>	<i>Cluster Shade</i>	<i>Cluster Prom</i>
1.7688E-04	3.9962E+02	-4.0964E+02	8.7709E-02	4.0363E+00	1.9196E+03	-3.3546E+01	2.2817E+05	1.2224E+08
1.3744E-02	1.0452E+03	-8.8027E+01	1.8795E-01	3.8417E+00	4.7442E+03	-9.4098E+02	-7.1590E+05	5.5199E+08
1.2864E-04	6.2188E+02	-3.9540E+02	6.6091E-02	4.1602E+00	2.2244E+03	-2.0033E+02	-2.8351E+04	1.3757E+08

Table 4 - Statistics generated from the images in Figure 6, after preprocessing

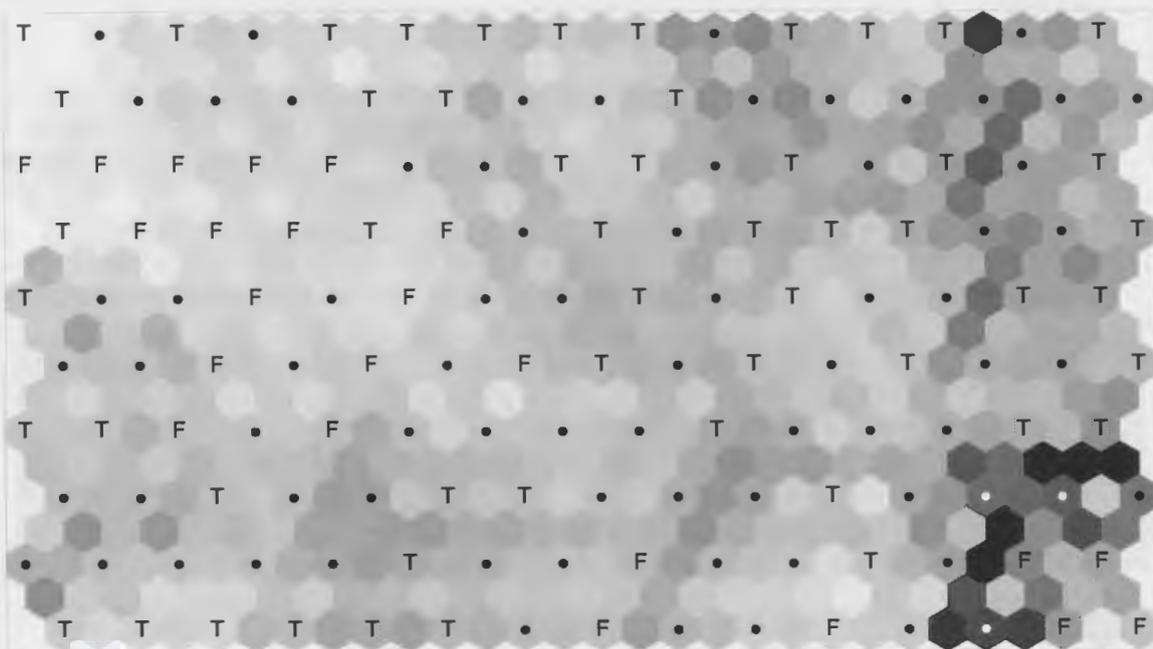


Figure 8 - Fingerprint and Baseline textures visualized

sets. A sample map is shown in Figure 11. Note that the statistics representing the composite textures are distinguished from those of the VisTex and NIST fingerprint sets. The four textures shown in Figure 10, for example, have been clustered near the center of the map. The composite textures tend to occupy the center of this example map, while fingerprints themselves are grouped to the left and VisTex textures are primarily found on the edges.

In addition to distinguishing between composite and non-composite texture statistics, the SOM was able to form sub-clusters among those composite texture statistics which were derived from the same class of base textures. For example, the map above has grouped together statistics representing the composite images involving tree bark (M0 and M22 near the center). Note, however, that these were not placed near the VisTex tree bark (T0, T2, T3 right of center).

8. Conclusion

We believe that self-organizing maps have value for visualizing GLCM texture descriptors. They show promise for both clustering and visualizing statistical vectors, encoding visual similarity, and recognition of textures not previously "seen" by the algorithm.

Future research should explore further the way that the SOM parameters – its dimensions and learning rate – affect the quality and usefulness of the map produced. One approach that has the potential to help with overall map quality is to use an 18 or 27 dimensional vector to represent each image, containing the original nine statistics computed on GLCMs with two or three different displacement vectors. This might be especially useful in dealing with textures that are rotated or skewed. It would also be worthwhile to explore the use of alternate texture descriptors, such as autocorrelation functions, run-length encoding descriptors, or wavelet statistics, to cap-

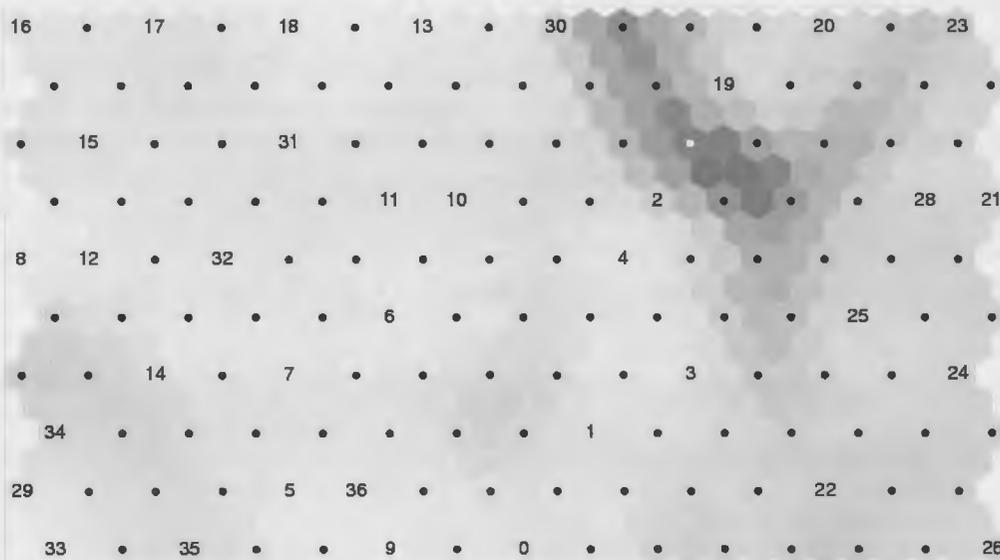
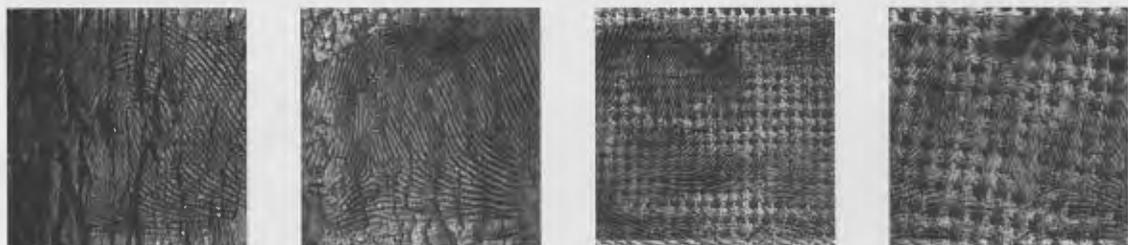


Figure 9 - Fingerprint image texture statistics visualized



A. Bark and fingerprint (M0) B. Bark and fingerprint (M22) C. Fabric and fingerprint (M7) D. Fabric and fingerprint (M8)

Figure 10 - Some of the merged textures and fingerprints

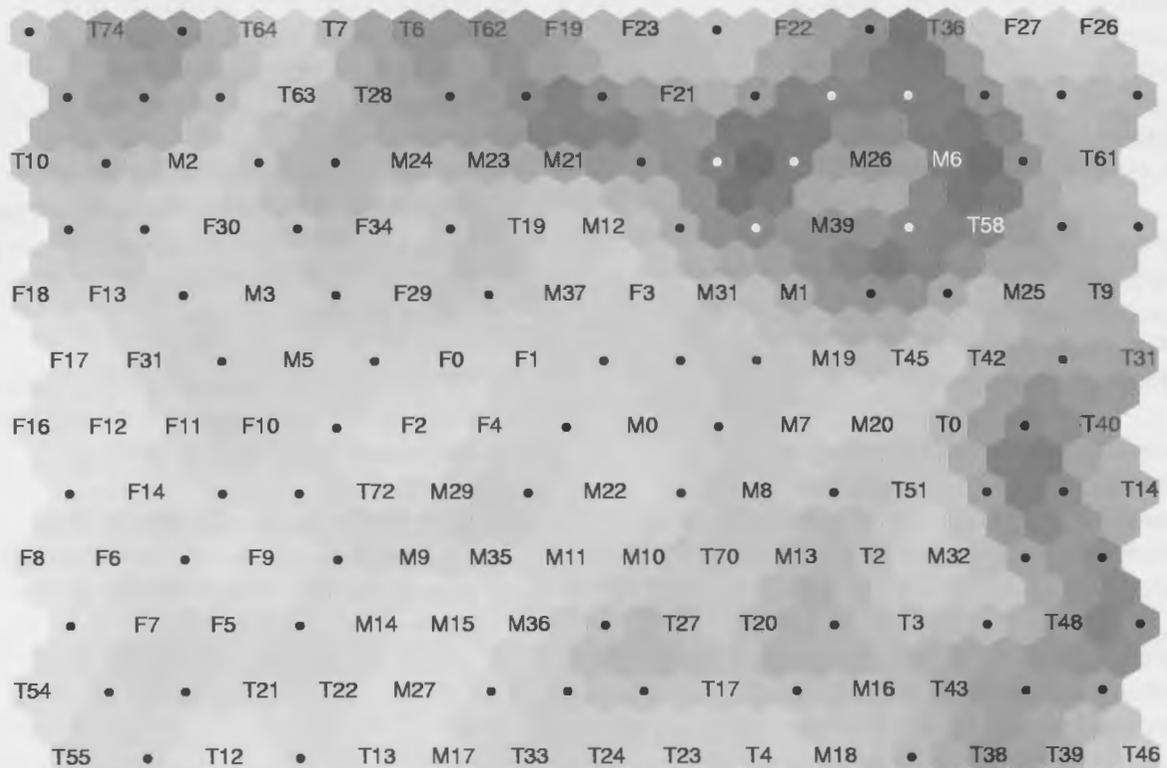


Figure 11 - Results from a map trained with fingerprint textures (F), VisTex textures (T), and clustered fingerprint/texture images (M). The numbers identify specific images in the database.

ture more of the texture's high- and low-level structure. We anticipate that this overlay could be especially useful in improving the visualization of vectors representing textures that are very visually similar to one another. We believe that this technique may also have forensic potential for recognizing the presence of surface fingerprints.

Footnotes

¹ The authors would like to express their gratitude to Indiana University South Bend for their support with a Student / Mentor Academic Research Team (SMART) fellowship.

² There are several parameters involved in this process – the dimensions of the map, and the length of the first and second training periods as well as the neighborhood radii and training rates. There are no theoretically solid methods for choosing these parameters, so parameters are determined empirically.

³ We realize that “visually similar” is imprecise; since visual texture is, to a large extent, based on human perception, and since we are dealing with common, well recognized textures, we felt this was reasonable.

Works Cited

1. Tamura, H., S. Mori, and Y. Yamawaki, “Textural Features Corresponding to Visual Perception,” *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-8**, pp. 460-473, 1978.
2. Sklansky, J., “Image Segmentation and Feature Extraction,” *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-8**, pp 237-247, 1978.
3. Tuceryan, M. and A. Jain “Texture Analysis,” *The Handbook of Pattern Recognition and Computer Vision (2nd edition)*, Chen et al., eds., pp. 207-248, World Scientific Publishing Co., 1998.
4. R. M. Haralick, K. Shanmugan, and I. H. Dinstein, “Textural features for image classification,” *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-3**, pp. 610–621, 1973

5. Kohonen, T., "Automatic Formation of Topological Maps of Patterns in a Self-Organizing System," Oja, E. and Simula, O., eds., *Proceedings of the Scandinavian Conference on Image Analysis*, pp. 214-220, Helsinki, Finland, 1981.
6. Kohonen, T., "Construction of Similarity Diagrams for Phonemes by a Self-Organizing Algorithm," *Technical Report TKK-F-A463*, Helsinki University of Technology, Espoo, Finland, 1981.
7. Kohonen, T., "Hierarchical Ordering of Vectorial Data in a Self-Organizing Process." *Report TKK-F-A461*, Helsinki University of Technology, Espoo, Finland, 1981.
8. Kohonen, T., "Self-Organized Formation of Generalized Topological Maps of Observations in a Physical System." *Report TKK-F-A450*, Helsinki University of Technology, Espoo, Finland, 1981.
9. VisTex Visual Texture Library, MIT Media Lab, <<http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>>
10. National Institute for Standards and Technology, Scientific and Technical Databases, <<http://www.nist.gov/srd/index.htm>>
11. Sharma, M., M. Markou and S. Singh, "Evaluation of Texture Methods for Image Analysis." PANN Research, Department of Computer Science, University of Exeter, United Kingdom.
12. XITE – X-based Image processing Tools and Environment - <<http://www.ifi.uio.no/forskning/grupper/dsb/Software/Xite/>>
13. SOM_PAK – the Self-Organizing Map Program Package - <<http://www.cis.hut.fi/research/som-research/nncr-programs.shtm>>



Figure 10 – represents vector file of textures split into sixteen sub-images

