

A society of models for video and image libraries

by R. W. Picard

The average person with a computer will soon have access to the world's collections of digital video and images. However, unlike text that can be alphabetized or numbers that can be ordered, image and video has no general language to aid in its organization. Tools that can "see" and "understand" the content of imagery are still in their infancy, but they are now at the point where they can provide substantial assistance to users in navigating through visual media. This paper describes new tools based on "vision texture" for modeling image and video. The focus of this research is the use of a society of low-level models for performing relatively high-level tasks, such as retrieval and annotation of image and video libraries. This paper surveys recent and present research in this fast-growing area.

Consider the following scenario about vision texture. Suppose you have a set of vacation photos of Paris and the surrounding countryside, and you accidentally drop them on the floor. Although they are out of order, you pick them up, sorting them back into two stacks—city and country. With only a quick glance at each photo, you are able to re-sort them to the right categories with high accuracy. How do you do this so quickly, without taking time to look at the precise content of each photo?

In this scenario, and many other picture recognition and sorting tasks, people appear to use relatively low-level information for making "quick glance" high-level decisions. Studies have shown that even pigeons with their pea-sized brains can discriminate images of water and trees¹ as well as impressionist and cubist paintings.² Inspired by these kinds of successful behavior, we have been exploring the use of collective low-level features, such as texture and color, for mak-

ing relatively high-level decisions about images. Such features tend to produce faster results than the traditional computer vision algorithms aimed at constructing detailed representations of everything in a picture. In this paper I describe several of the models we have explored, and the important additional step of combining them into systems that interact with humans.

Vision texture

Consider a computer solution to the scenario previously described. A simple measure of local orientation over scale, a low-level operation designed to mimic part of what scientists believe occurs in the human visual system³⁻⁵ was used with some simple decision rules for classifying a set of 98 vacation photos. Based on only a quick decision with the low-level orientation information, 91 out of 98 of the photos were correctly classified into the categories "city or suburb" or "other."⁶ Two of these photos are shown in Figure 1.⁷ The careful use of low-level collective properties of image data for relatively high-level visual tasks is referred to as *vision texture*.

Low-level features such as color and texture are not just for low-level tasks. Although vision texture is not sufficient for completing high-level relational tasks such as "find an image with an oak tree on the left and a lake on the right," there are numerous demonstra-

©Copyright 1996 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

Figure 1 Quick glance recognition: city or country?



tions of the success of vision texture for achieving or helping achieve relatively high-level tasks. Swain and Ballard⁸ illustrated the use of simple color histograms for retrieving images from a diverse database, and Syeda-Mahmood has shown how a combination of color and texture features can speed up selection of items of interest in photos.⁹ Texture has also been shown to be powerful for recognition of motions.¹⁰

Texture: beyond the traditional definition. There is much more texture in the world than most people realize. Texture is ubiquitous; it is felt on the tiny surface of a shriveled pea, can be heard in the interwoven melodies of a fugue, can be seen in the rocking motion of a boat, and even shows up in human affect and behavior patterns. Eluding precise definition, texture is usually distinguished by being tactile, patterned, rhythmic, or noisy.

It is generally an ill-posed problem to say “find the texture in this picture.” Texture eludes precise definition. Some researchers define it like pornography, “you know it when you see it.” I find it helpful to list properties usually associated with texture, such as the

three that follow. These three properties are not mutually exclusive, but are separated for easier discussion of how they influence applications.

Property 1: lack of specific complexity. The first property is illustrated by considering three categories of patterns, shown by the one-dimensional (1D) strings of letters in Figure 2. (These strings were inspired by the discussion of different kinds of entropy in Reference 11.)

The first string (String 1) is a 1D periodic texture. It has a basic primitive, a specific set of rules for replication of the primitive, and allowance for minor perturbations. The primitive may be complex, but its complexity is leveraged over the whole pattern, resulting in low overall complexity as the string becomes longer. Periodic textures like this show up in physical materials such as nylon and crystals, and in audio segments such as the sound of a copy machine repeatedly sounding “ker-chunk ker-chunk slurp,...” Periodic textures also occur in two-dimensional (2D) imagery of tile floors, and in repetitive space-time patterns such as the two feet of a person riding a bicycle.

Figure 2 Defining texture: two texture strings and a third string that is not a texture

HiccupHiccupHiiccup
sh ysSTehtignaSio m
This Says Something

The second string (String 2) is a sample of a 1D stochastic texture, perhaps generated with a random number generator or filtered noise. A random sequence may look complex, but it has no specific order; it is characterized by a probability distribution. Random polymers, the sound of applause, and nucleic acids are other 1D examples; turbulent water and the footsteps of children playing tag make higher-dimensional stochastic textures.

The third string (String 3), like the structure of DNA (deoxyribonucleic acid) and proteins, is distinguished by having both specific order and complexity. Although it is an anagram of the second, and may be extracted from the same probability distribution, its specificity makes it qualitatively different. (Note that Shannon deliberately left “meaning” out of his probability-based information theory.¹²) This third string and its higher-dimensional analogues are not textures. For example, an analogous image would be a human face; without its underlying specific arrangement of eyes, nose, and mouth, it would cease to be recognized as a face. A single face is not a texture.

Note that my use of “texture” here includes most textures used in computer graphics, but is not as broad as the use of the term “texture” in the “texture-mapping” literature. The latter refers to arbitrary pixel maps placed over a three-dimensional (3D) structure to add realism to the scene. In computer graphics, an image of a face might be “texture mapped” onto polygons or a finite-element mesh to render a more realistic 3D face. The face is not a texture by the properties outlined here, but is being treated like a texture with respect to the surface onto which it is being mapped. Similarly, a texture image such as sand might be texture-mapped onto a 3D polygon shaped like a mound to render the effect of a 3D pile of sand.

The three strings may also be combined in higher dimensions. For example, an image of a plowed field combines randomness along one direction with peri-

odicity along the other. The Wold model, which will be highlighted later, is based on such a separation of random and deterministic components.

There is no hard boundary between the three cases. Consider the following two examples:

Example 1: String 3 can be replicated, resulting in a periodic texture like the case of String 1. The boundary between nontexture and texture is analogous to the boundary between nouns used for counting and nouns used to describe mass: Asking how many replications of a nontexture it takes to make a texture is like asking how many grains of sand it takes to make a pile.

Example 2: String 3 can be gradually permuted until the order is no longer recognizable as a meaningful sentence, and it becomes like the case of String 2. An analogy in the image domain would be to overlay multiple views of a face, so that it suddenly had multiple eyes, noses, and mouths, no longer in the expected specific arrangement. The result is an effect similar to that achieved by Picasso with cubism, and may explain why people (and pigeons, perhaps) sometimes think such paintings look like textures.

Property 2: high frequencies. Although both texture and nontexture can contain high-frequency changes, these tend to occur more with texture. This property is perhaps most important, and annoying, to researchers in image coding where standardized coding methods utilize basis-functions such as the discrete cosine transform. These methods attain the best compression in smooth (low-frequency content) areas, so that pictures with lots of texture tend to be hard to compress efficiently.

Note that extreme smoothness can still be considered to be a texture, especially in the tactile domain (“feel the ‘silky smooth’ texture of this garment”) but in digital imagery, smooth regions generally are considered as nontextured.

Property 3: restricted range of scale. Textures, unless they are truly fractal,¹³ tend to exist over a finite range of scales. Tree bark may look smooth from a distance, grooved as you move in closer, and pitted when you press your nose to the trunk. A brick wall looks periodic from a distance, but loses its periodicity when you are so close that you can see only a few bricks. This lack of persistence of texture over scale complicates the association of objects with texture; a range

of scale and “typical views” must be a part of the association.

Scaling similarity also shows up in a less obvious way—across very different phenomena at different scales. In his delightful book on patterns in nature, Stevens¹⁴ shows pictures of gas clouds and of milk poured into a black slate sink—two different materials at scales ranging from a centimeter to over ten quintillion kilometers, both of which can be generated as “turbulence” textures. Stevens examines many of the common behaviors of natural patterns, including close packing, spirals, branching, shrinking surfaces, and turbulence—revealing a small number of underlying mechanisms responsible for an astronomical variety of patterns. This variety of mechanisms for forming patterns in nature suggests that we might find more than one model useful in forming digital patterns.

The three properties just described—lack of specific complexity, presence of high frequencies, and restricted scale—hint at the difficulty of characterizing textures, but more importantly, illustrate an expanse of possible forms. Texture occurs in audio, chemical structures, motion, imagery, and even human behavior patterns. A significant research challenge is to develop a family of models useful for representing, manipulating, comparing, and recognizing textures in digital libraries.

In the rest of this paper, the focus will be on texture models for image and video and on the systems we have developed using vision texture for applications such as browsing, retrieval, and annotation.

A society of models

A ski jumper “shoots” out of the starting gate, speeds down the snowy slope, forms an airfoil—flying—steady—then lands. To predict the jumper’s motion, one might picture a straight trajectory lifting at the top of the hill, lowering at the bottom, and followed by a switch into two possibilities at the instant of landing. At that instant, the predictor may switch from a “straight-ahead” model, to a “tumbling-out-of-control” model. Two models—straight, or random—are useful for efficiently describing the motion. Similarly in football, whether we watch the motion of the ball being passed, carried, or fumbled, we switch naturally between different mental models of prediction. The right repertoire of models, and their proper combination, is more effective than trying to use one model for all tasks.

Figure 3 contains several models that have been used in computer vision, image processing, and computer graphics. Some of these are general enough to represent arbitrary signals and may be used for synthesizing data—perhaps for simultaneous compression and recognition in digital libraries. Other models only capture some features of a given signal that are useful for recognition or query. “Analysis” usually refers to the estimation of features or parameters of the model. Sometimes model features might be used (for example, within an optimization framework) to approximate a reconstruction to the data, but in general they need not be sufficient for reconstructing the data. Such features might be useful, however, for discriminating among several categories of data. Both kinds of models—those that can resynthesize the data, and those that cannot, have applications in digital libraries.

One of the realities of research is that each model tends to have a trendy period of use, and then it is abandoned in pursuit of a presumably newer and better model. Instead of searching for one “best” model, the approach here is that it is important to study a variety of models, to learn what they do best, and to learn how they may be effectively combined. This approach shares the spirit of Minsky’s *Society of Mind*,¹⁵ whereby specialized agents, or models in this case, interact to make sense of what they see. Just because a model is capable of representing everything does not mean that it is the best one to use for everything.

In the rest of this section I will survey six models that have been the focus of our recent research. These six models are chosen to represent a variety of forms, including deterministic, stochastic, mixed, linear, and nonlinear forms. Some have parameters that are physically motivated, some that are perceptual, and some that are semantic. Most can be applied to arbitrary digital signals, although the emphasis here is on modeling imagery in space and time. Information on the other models in Figure 3 can be found in the references, especially overviews such as Reference 16 and Reference 17. There is not space here for equations and details, but these are referenced for each model. The focus in the descriptions that follow is to familiarize the reader with each model, highlight some apparent strengths and weaknesses of each model, and point to important relations between the models. A following section of this paper then discusses two systems we have built that rely on a society of models for more effective performance.

Figure 3 A society of models. Although some of these can model any signal, each has different strengths and weaknesses.

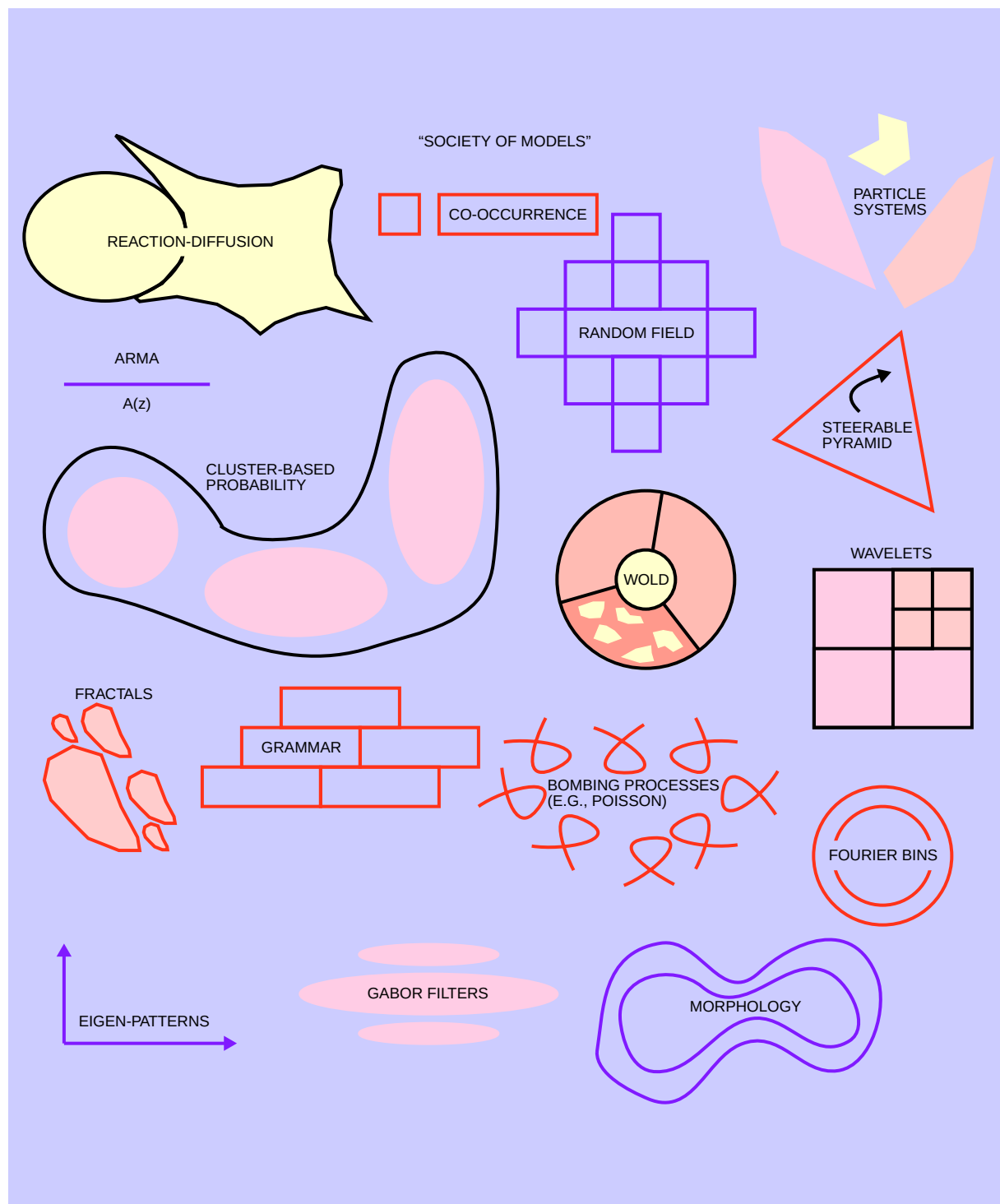


Figure 4 Biological patterns like these can be mimicked by digital texture models.



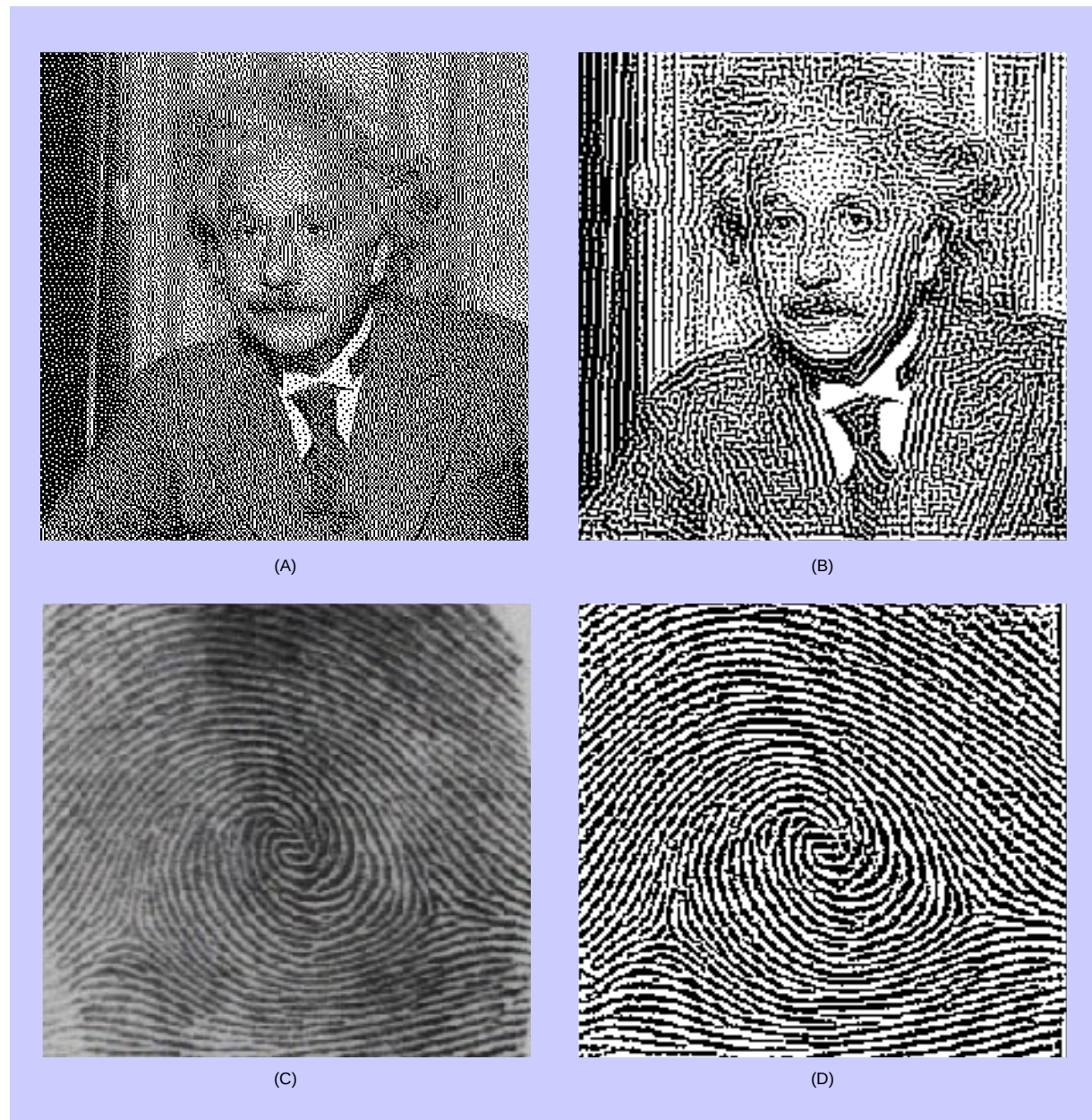
Reaction-diffusion models: beyond zebra stripes and leopard spots. Nature appears to use simple non-linear mechanisms for pattern formation, or *morphogenesis*. For example, butterfly wings exhibit a great variety of patterns, all of which must be produced within a simple, light-weight, insect structure. The spots and stripes on lepidoptera are also found on brightly-colored tropical fish, zebras, leopards, tigers, cheetahs, birds, and more. In a digital library of such imagery, one might expect a reaction-diffusion model to be powerful for both representation and retrieval. The painting by Lenore Ramm in Figure 4 illustrates some of the variety of animal patterns that are well modeled by digital texture models. In particular, reaction-diffusion models may be used for efficient

description of most natural patterns involving spots and stripes.

Turing proposed in 1952¹⁸ that dappled patterns could be synthesized by a set of coupled nonlinear partial differential equations known as a *reaction-diffusion* system. Under certain conditions, reaction-diffusion models also can be used for analysis.¹⁹ Inspired by Turing's work, we have developed a new nonlinear "M-Lattice" model that solves the biggest practical problem of the original Turing model (boundedness), and is still good at making spots and stripes.

Figure 5 demonstrates an application to halftoning, the representation of gray-level images by black spots

Figure 5 Use of a reaction-diffusion model (stripes and spots) for halftone images



on a white background.²⁰ In Figure 5, a reaction-diffusion model was used to make a “faithful” halftone in Figure 5A, a “special effects” halftone in 5B, and, using an original noisy fingerprint image (5C), a binary enhanced fingerprint image in 5D. The new M-Lattice solves a variety of nonlinear optimization

problems, such as the creation of the “Wall Street Journal style” halftone, that grows patterns along visually dominant directions, much like the hand-drawn versions made by artists.²¹ The basic idea here is that the error introduced by halftoning gets pushed into perceptually favorable directions, along lines that

already exist in the image. We have also demonstrated the creation of color halftones with these effects using the M-Lattice.²²

Stripes, such as those on zebras and fish, are well-modeled by the nonlinear Turing and corresponding M-Lattice models. However, digital libraries of zebras and fish are not presently as abundant as those of fingerprints. Human fingerprints, which resemble bifurcating stripes on zebras, have recently been successfully modeled with the new M-Lattice for the purposes of enhancement and binarization. Instead of merely removing noise, the M-Lattice boosts the underlying fingerprint pattern, effectively suppressing unwanted noise and intensity variations.²⁰

The reaction-diffusion model has found applications in image processing,^{20,23} computer vision,²⁴ and computer graphics.^{25,26} In the latter, the emphasis has been on synthesis, although the synthesizing parameters could certainly be stored in a database of synthetic imagery and used for data manipulation, annotation, and retrieval. The effectiveness of reaction-diffusion as a biological model, not just for animal coat pattern formation, but also for emergence of structure of all kinds, is an ongoing research topic in mathematical biology.²⁷ In the digital arena, the model has been most successful in the synthesis of textures or images comprised of spots and stripes. However, the model is still new and largely unexplored. As a nonlinear model with a huge space of possible behaviors, it will be some time before its strengths and weaknesses are fully characterized.

Markov random field models: from grass and sand to monkey fur. The reaction-diffusion model is deterministic. However, there is another class of models that bears a resemblance to reaction-diffusion but which is stochastic—the class of Markov random field (MRF) models. Unlike most texture models, an MRF is capable of generating random, regular, and even highly structured patterns. In theory, it can produce any pattern. It does not just describe some characteristics for distinguishing textures, but it can be used for both texture analysis and synthesis.

The MRF has simultaneous roots in the Gibbs distribution of statistical mechanics and the Markov models of probability. The Gibbs distribution has a rich history of applications in physics including the modeling of lattice gases, molecular interactions in magnets, and ordering processes in condensed matter. In computer vision and image processing, the MRF is touted

for its ability to relate the Markov conditional probabilities to the Gibbs joint probability. It can be easily incorporated into a Bayesian framework, making it flexible for a variety of applications.

Hassner and Sklansky²⁸ appear to have been the first to suggest the use of Markov and Gibbs models for image texture. Cross and Jain²⁹ conducted the first explorations of the MRF for gray-level texture modeling and showed that it generated natural-appearing microtextures such as grass or sand. A Gaussian MRF has been applied to texture classification and modeling by Chellappa and Chatterjee^{30,31} and Cohen et al.³²

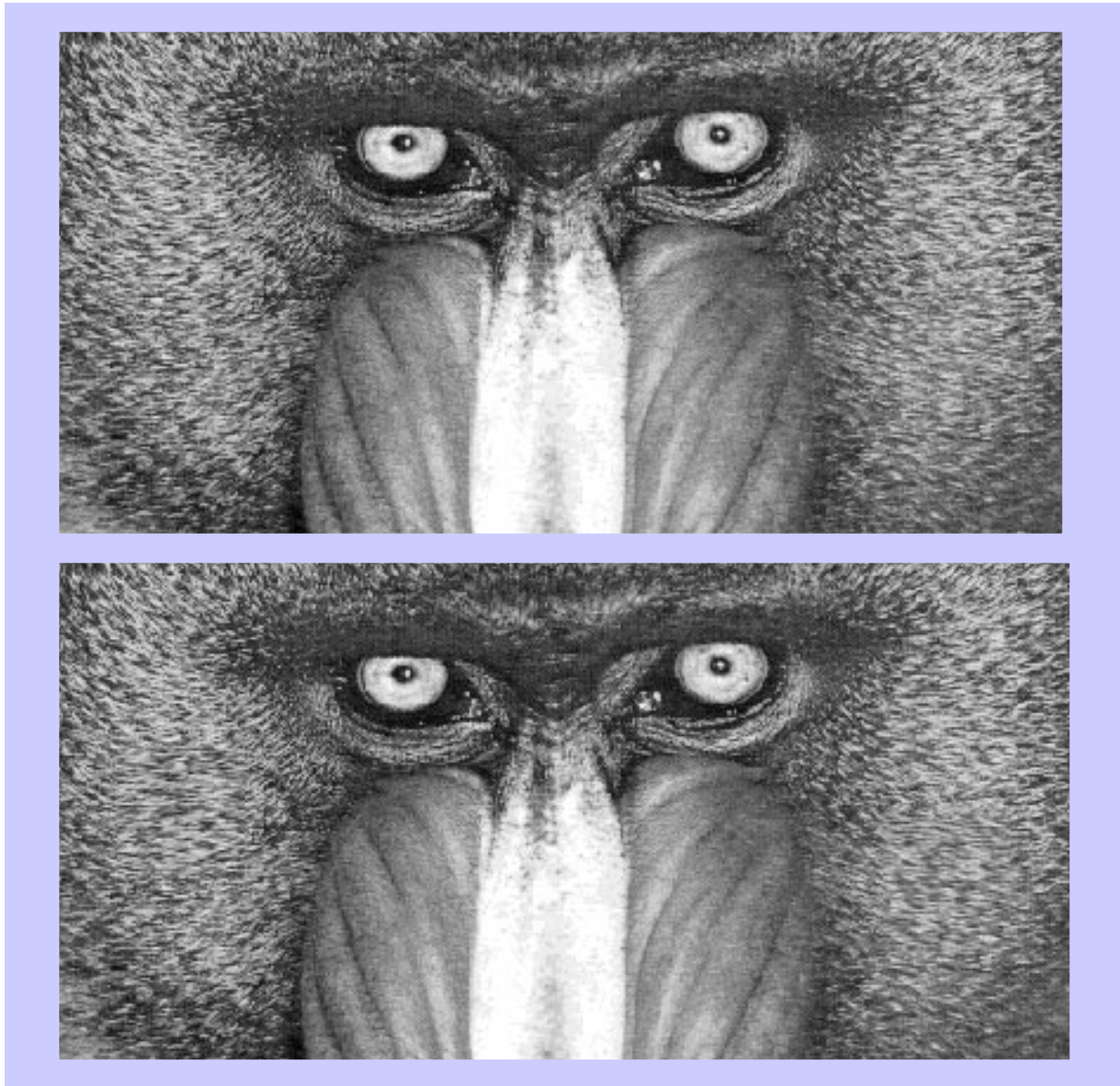
Given successful use in these small sets of data, the MRF should also be useful in large digital library problems, when the library data are well-described by the model. For example, the aura framework derived from an MRF model has been shown to be useful for characterizing spatial yields of semiconductor wafers.³³ Searches through a database of wafer-yield imagery might therefore favor this model for finding similar patterns.

The interplay between microscopic dynamics and macroscopic force, such as that associated with a phase transition³⁴ triggered by temperature, is an important factor in natural pattern formation. The effects of a temperature parameter on pattern formation with MRFs have been studied³⁵ revealing relationships between structuring models within mathematical morphology and the useful statistical features of co-occurrence.³⁶ However, these relationships also indicate limitations on the patterns that can occur at low temperature.³⁷ Although in theory the MRF can model anything, these low-temperature relationships point to weaknesses of the MRF model.

In particular, although the MRF can make structures such as the stripes and spots favored by the reaction-diffusion model, it does not typically make such patterns unless coupled with an external structuring force, or forced into a low-temperature state.³⁸ For example, running at low temperature on low-frequency structural cloud images was successful at simultaneously capturing cloud texture while preserving cloud shape.³⁹ In general, the expertise of the MRF does not seem to lie in large-scale structured patterns, except in a few special cases, and when careful temperature control is exercised.

The strength of the MRF appears to lie with homogeneous microtextures and simple attractive-repulsive

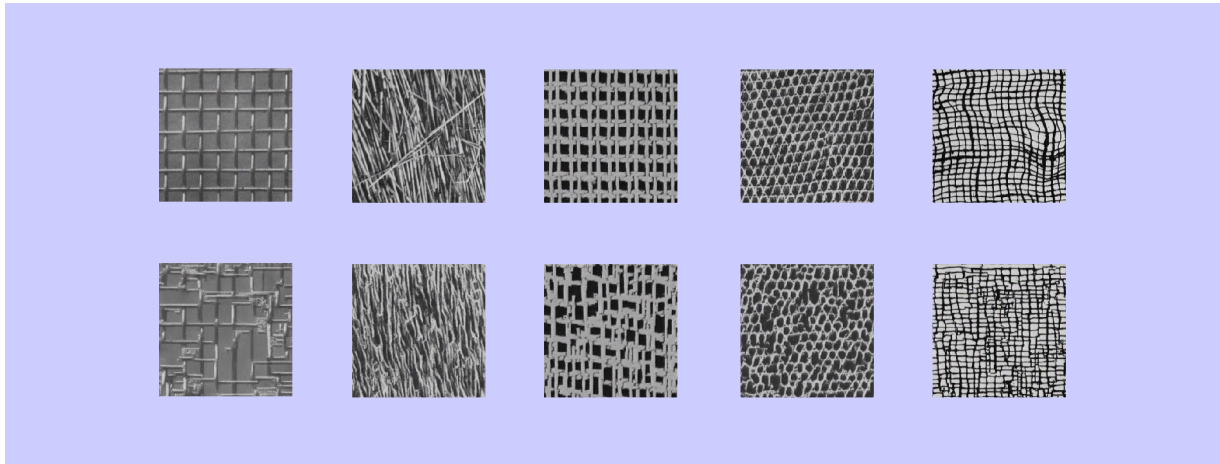
Figure 6 Illustration of a strength of the MRF model. The top image is the 256 by 512 original; the bottom image is the same except for two 64 by 64 patches of synthetic fur. Can you see them?



interactions. Figure 6 shows the use of an MRF model for synthesizing the microtexture of fur in two patches of a mandrill image. Details as to how this was done, as well as its potential for model-based semantic image compression, are discussed in Reference 40. Although the model is successful for fur in this example, the reader should keep in mind that the model is

not typically successful on nonhomogeneous or non-microtextures, and was not found to be successful when trained on other parts of the mandrill image. To summarize: in theory the MRF can represent all patterns; however, in practice, its strengths make it suitable to only certain kinds of imagery that might occur in a digital library. Like all the models we have exam-

Figure 7 Top row: 256 by 256 patches used to train cluster-based probability models. Bottom row: deterministic multiresolution synthesis. The textures are, from left to right, D1 (aluminum wire mesh), D15 (straw), D20 (magnified French canvas), D22 (reptile skin), and D103 (loose burlap).



ined, its utility depends greatly on the contents of the digital library.

Cluster-based probability modeling: audiovisual patterns. As previously mentioned, the MRF can theoretically represent any pattern, but is typically only good at capturing low-order interactions due to the complexity of its parameter estimation. The mandrill fur is a typical example of what it is good at synthesizing. The MRF fails at capturing patterns like those shown in the top row of Figure 7 (from the Brodatz⁴¹ album). To capture more complicated structures than in microtextures, we now consider a higher-order probabilistic model.

The key problem with increasing the order of a probabilistic model is that it exponentially increases the space of possibilities. For example, to consider joint interactions among a set of 14 pixels in a 256 gray-level image results in 2^{112} possibilities. This number dwarfs even the total number of images all of humankind could have ever seen, a mere 2^{70} possibilities. (The latter assumes 10 billion humans with their eyes open 24 hours per day, watching 30 frames per second, living 100 years each.) Clearly, a model dealing with this many possibilities will run into practical problems.

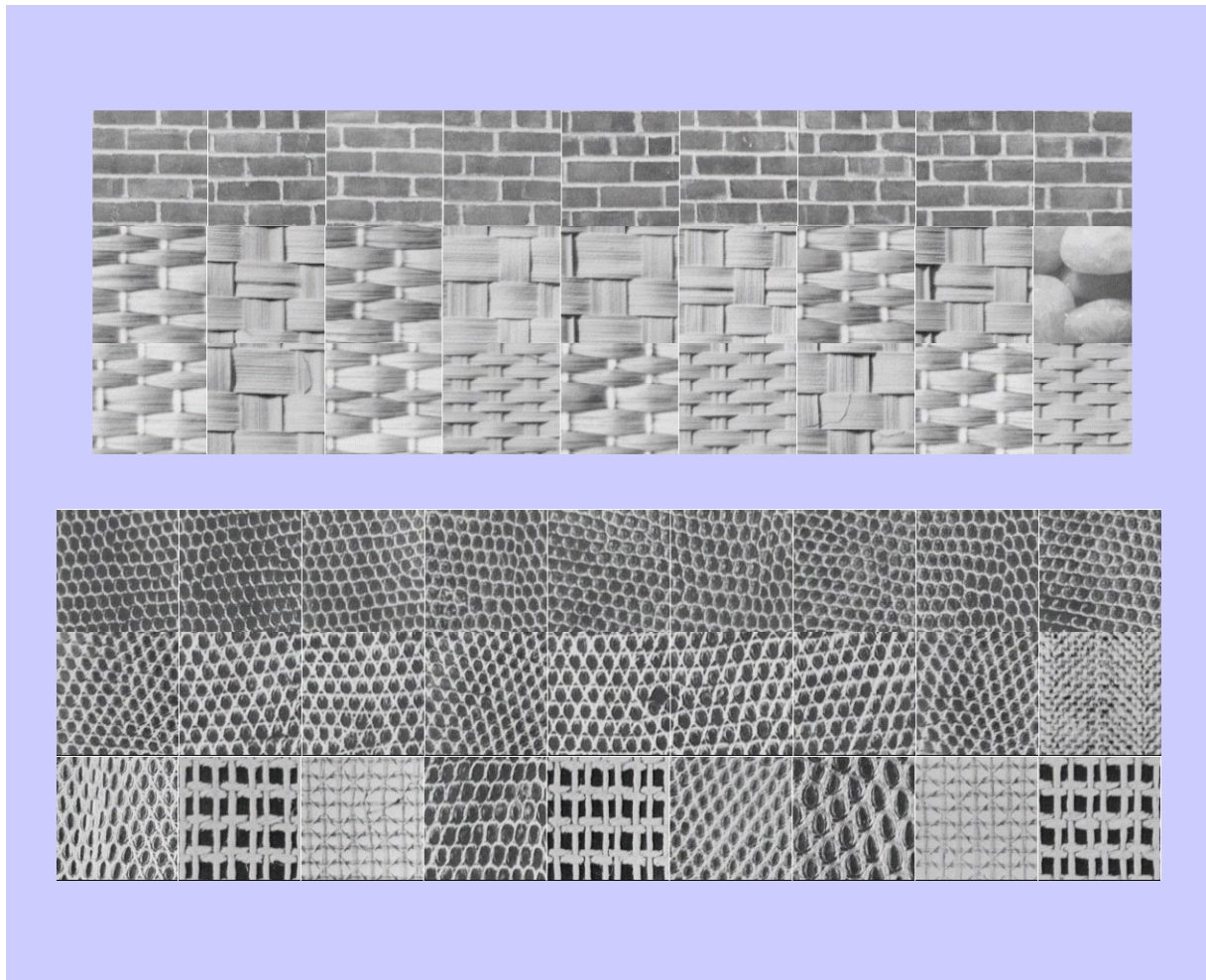
The approach taken to make this model practical is described in Reference 42. To illustrate its power at capturing both microtexture features and higher-struc-

tured features, its parameters have been trained on five patterns shown in Figure 7, using 14th order joint probability statistics. To jointly model 14 variables is a significant increase over the MRF; the latter is computationally tractable usually only for up to 3rd-order joint statistics. A multiresolution maximum-likelihood method was used to synthesize textures from the model parameters; these results are shown in the bottom row of Figure 7. Notice that the probability distributions did not involve enough variables to enforce globally regular structures; nonetheless, much of the character of the original is present in the full-resolution result. For example, the probabilistic model trained on the wire mesh in the first column captures relatively high-level features such as shading, bending, and even occlusion of the wire strands.

The cluster-based probability model implemented here is related to several other models, such as Gaussian mixture models; these relations, along with the application of this model to image restoration and compression, are discussed further in Reference 43. One of the drawbacks of the model is that it presently requires a lot of parameters compared to other texture models. Research is under way to determine how the parameters can be leveraged across large classes of patterns, to make the model more efficient for use in digital libraries.

The cluster-based probability model has recently been shown to be capable of realistic sound texture synthe-

Figure 8 Two examples of using Wold features for pattern retrieval, searching for patterns similar to the pattern at upper left



sis,⁴⁴ and to perform well on certain perceptual similarity comparisons of sounds.⁴⁵ Indeed, a truly effective society of models will include models that work not just for visual features, but also for arbitrary perceptual and semantic information features. Digital libraries often contain mixed media such as audio and image; models that can handle multiple media offer savings in design time, development time, and overall system cost.

A new Wold model for perceptual pattern matching. What features are important to people when measuring similarity in pictures? A perceptual study by Rao and Lohse⁴⁶ has shown that the top three features

may be described by (1) periodicity, (2) directionality, and (3) randomness. A model that explicitly gives control over these features would potentially provide more perceptual control over pattern formation and visual queries.

In statistics, there is a theorem by Wold that provides for the decomposition of regular 1D stochastic processes into mutually orthogonal deterministic and stochastic components. For images, this results in a decomposition into three components, which approximately correspond to periodicity, directionality, and randomness. As such, the Wold model is one of the few models that has intuitive parameters, or semantic

“control knobs.” An implementation of this model for analysis and synthesis of homogeneous textures can be found in Reference 47.

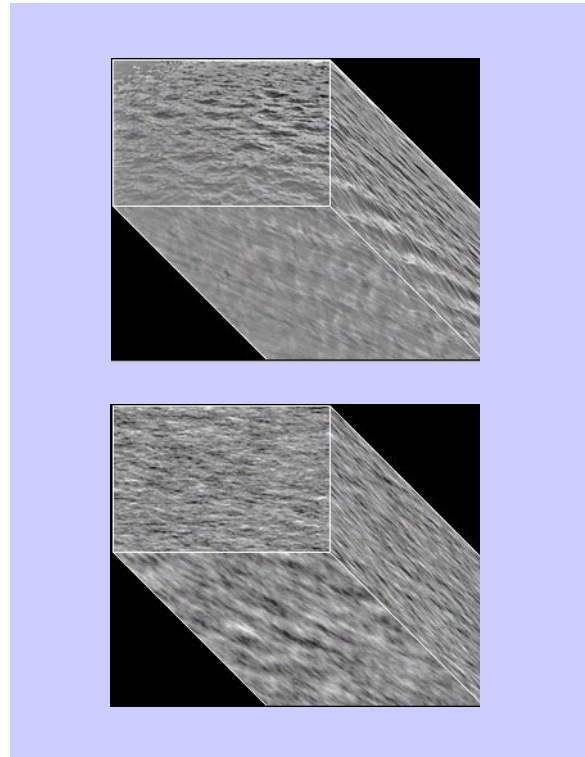
For the purposes of image retrieval, we have developed a new implementation of the Wold model. This implementation facilitates the finding of perceptually similar patterns in a database containing both homogeneous and nonhomogeneous textured images.⁴⁸ When the user selects a given image, similar-looking images are retrieved. Examples are shown in Figure 8. The upper left image in each of the two parts of the figure is the one selected by the user. The images that follow represent the closest images in raster-scan order from the selected image. Although the images here are from the Brodatz database, they could just as well be from a large database of fabrics, tiles, wall-coverings, and other textiles, facilitating searches by consumers and designers.

Although the Wold model was found to be the most successful of five texture models⁴⁸ for retrieval in the Brodatz database, it is not necessarily the best for an arbitrary set of imagery. To summarize, its strengths appear to lie in natural pattern similarity, especially when periodicity, directionality, and randomness are distinguishing features. One of the weaknesses can be seen in the second row of Figure 8, in the right-most image, where round stones were retrieved, due largely to the presence of high contrast horizontal edges near the center of this image.

Stochastic model for temporal textures. Video is full of motion, providing a new challenge for texture models. Some motions are rigid, like a car moving across a scene, and can be captured by simple non-textural models. However, motions such as blowing leaves and wavy water are nonrigid, and require models that exploit local collective properties—temporal texture models.

Temporal texture is a relatively new research area; only in the last few years have researchers been able to deal with the growth in computational complexity and storage caused by an extra dimension of raw data. Our work in this area has focused on treating video as a spatio-temporal image volume. Patterns in the volume show up as a result of periodic or random motions—for example, a person walking across a scene results in a periodic braided pattern at leg-level.⁴⁹ The types of queries we hope to address with this research are queries such as “find scenes with moving water,” or “are there other scenes where a per-

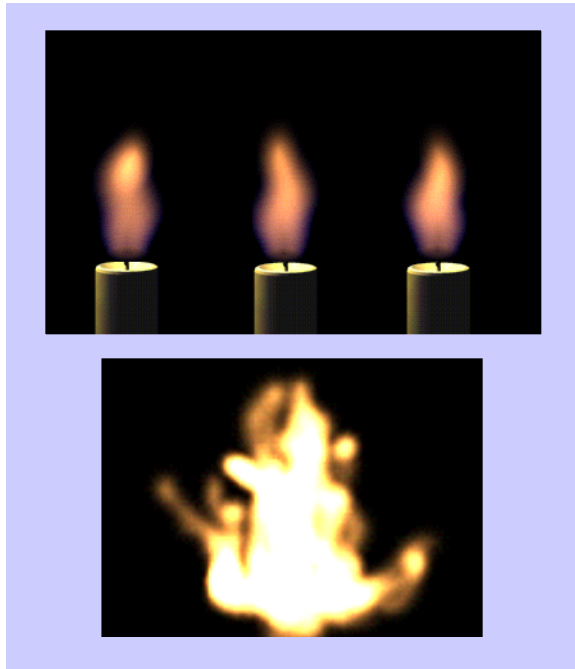
Figure 9 Top: original river video sequence. Bottom: synthetic sequence made from STAR model, illustrating similarity to the original.



son is walking?” Like spatial texture, temporal texture will need to be augmented with other information before it can address relational queries such as “find dogs chasing cars.”

In an effort to first formulate a general temporal texture model, a linear auto-regressive model (of the auto-regressive moving average [ARMA] family in Figure 3)⁵⁰ was extended for stochastic temporal textures. The standard 2D model was augmented to form a linear spatio-temporal auto-regressive (STAR) model, which predicts new image values based on a volume of values lagged in space and time.⁵¹ Using the STAR model, parameters for stochastic temporal textures were estimated, and the motions were resynthesized from the parameters. Resynthesis of motion textures such as steam, river water, and boiling water were found to look natural. These patterns might be thought of as temporal microtextures in that their perceptual characteristics are well-captured by pair-wise (2nd order) statistics over a small volume of the data.

Figure 10 Synthetic flames



An “x-y-t” volume of an original river sequence and a synthetic river sequence are shown in Figure 9, showing how perceptually similar they appear even though the one at the bottom was synthesized from model parameters. Although the STAR model was found to be strong at characterizing such homogeneous temporal textures, it was not found to be able to capture the structure in less homogeneous temporal patterns, such as swirling water going down a drain. Such patterns, like their spatial counterparts, seem to require either a larger joint interpixel characterization, or coupling with some global structure, as provided by the MRF external field. Nonetheless, a digital library might contain data for which the STAR model is the best choice. Alternatively, a model that directly incorporates mechanisms of swirling and other fluid motions might be better for some types of queries. The sixth model, described next, is an example of a model with explicit physically motivated mechanisms to control motion behavior.

Synthetic flames via polygonal particle systems.

One of the most challenging temporal textures to model is fire. Fire is one of nature’s greatest actors, able to evoke a wide range of feelings through its emotional and destructive power. For filmmaking, fire

is extremely difficult to control, and results in the expensive construction and subsequent destruction of objects on the set. Valuable resources are spent trying to exploit the power of fire through pyrotechnic techniques, and ultimately the range of available effects is limited by the laws of physics.

We have developed an interactive model for synthesizing fires that look real, respond properly to wind and gravity, light their environment, and spread over and char 3D objects.⁵² The flames are rendered using a technique based on modified particle systems. Each particle is a shaded translucent polygon, which combines with others to build the flickering flames. The flames are coupled with a physically based spreading mechanism to achieve realistic movement around polygonal 3D objects. The model parameters were designed to give graphic engineers semantic control knobs to change factors such as flammability of the underlying material or velocity of the wind, and to have the fire respond in the expected natural way. The resulting model makes it easier for realistic-looking synthetic fires to be placed into both artificial and natural scenes. Hence, the model avoids the costs and dangers associated with real fires, while giving a greater possible variety of effects. Additionally, the parameters can be set to control flame density, shape, blending, and noisiness, allowing nonphysical special effects. A few flames are shown in Figure 10.

Models with semantic parameters such as this flame model have a variety of uses beyond synthesis. As designers construct digital libraries of synthetic video and graphics, it becomes useful to use synthesis parameters for retrieval: “what was the name of that file that contained flames blowing in the wind but not spreading?” It is also possible that some of the parameters of the synthetic models might be estimated from natural footage, given that the parameters are physically motivated; this is an unexplored research area. Currently, the model parameters also allow fast and easy manipulation, so that a user may craft a variety of fires (candle flame, roof fire, etc.) either for modifying a particular retrieved scene to be closer to what the user wants, or for creating a prototype to search on. An example is “find fires spreading up vertical structures.”

Systems for browsing, retrieval, and annotation

The six models previously highlighted do not solve all the problems in texture modeling, muchless all the

problems in digital libraries. However, they illustrate a variety of areas of expertise, which can work individually or collectively to assist in representing, manipulating, comparing, recognizing, and annotating data in digital libraries. Which one or ones should be used? Unless a digital library contains a highly restricted set of data, which is known in advance, we cannot expect a single model to be best at all the tasks demanded of the digital library. A model that is good for finding flames that move in a particular way is probably not going to be good for finding particular human motions. In this section I briefly survey two systems we have built that incorporate vision texture and a society of models for assisting in browsing, retrieval, and annotation of image and video. Both systems are flexible in their abilities to incorporate a variety of models. The first depends on the user to select the models; the second learns to select or combine models automatically.

Photobook: browsing and retrieval. Both academic and industrial scientists have begun researching and developing systems to assist users in navigating through digital imagery. Some of the earliest and largest research efforts have been at IBM Almaden,⁵³ the Institute of Systems Science (ISS),⁵⁴ and MIT.⁵⁵ Early results have already been made into products, and can be explored interactively on the World Wide Web.^{56,57}

The first system developed at the MIT Media Laboratory was Photobook. Photobook is an interface that displays still images and video keyframes, and offers access to a variety of tools for browsing and retrieval. Photobook currently interfaces to databases including faces, animals, artwork, tools, fabric samples, brain-ventricles, and vacation photos. Depending on the category of images, different algorithms are available for assisting in retrieval. Each image has precomputed (off-line) features associated with it, so that when a user selects an image of interest, the system instantly updates the screen showing other images in the database most similar to the selected image.

The problems of what models to use for image representation and how to measure image similarity are challenging research problems for the image processing community.⁵⁸ Photobook, like the systems of References 56 and 57, allows the user to select manually from a variety of models and associated feature combinations. As a research tool, Photobook assists in rapid benchmarking of new pattern recognition and computer vision algorithms. An example interaction with Photobook, looking at video keyframes, is shown

in Figure 11. The query is “Are there any images similar to the image of the violin player shown at the top left?” After searching a database of several hundred video keyframes, the result is the series of images shown in the figure, ranked by similarity to the query image in terms of their visual content.

Experience interacting with the Photobook system has taught us that although it saves time in browsing and retrieval tasks, the job of selecting which model to use, or which combination of features for searching is generally nonintuitive. Although an expert who works with the models can learn which of them tend to work best on which data, this kind of expertise only holds across uniform databases, such as fingerprint images or face images. For general consumer photos, stock photos, or clip-art services, there may not be one winning model or fixed combination of models, but these may need to vary within the database, or vary with each new search. Even the expert with good intuitive understanding of the features rapidly becomes frustrated at how often the settings to combine features need to be changed for optimal performance.

The model combination in Photobook and similar industrial systems is feature-based, and tends to be limited to linear combinations of features—e.g., “Use 60 percent of texture model A, 20 percent of texture model D, 10 percent of color model B, and 10 percent of shape model A.” Unfortunately, users do not naturally sort images by similarity using this kind of language. In particular, as the dimensionality (based on total number of model features) increases, intuition about how to pick relative weightings among features is lost. The need to determine all the weightings for multiple features, and hence for the society of models, is a problem that plagues all existing retrieval systems to date. A solution to this harder problem was a key motivation for the system described next.

FourEyes: learning from user interaction. People have different goals when they interact with a digital library retrieval system. Even if they are nominally interested only in annotation, or only in retrieval, they are likely to have different criteria for the labels they would give images and the associations they would like retrieved. These criteria tend to be data-dependent, goal-dependent, culture-dependent, and even mood-dependent. On top of this unpredictability, the average user has no idea how to set all the system knobs to provide the right balance of color, texture, shape, and other model features to retrieve the desired data.

Figure 11 Photobook vision-based content query: a series of images selected based on the original in the top left



A society of models is most powerful when the models are well-matched to the problems, where the problems may depend significantly not just on the data, but on the present user's notion of similarity. The request "find more examples *like this*" has many right answers, and different models or model combinations may perform best for different answers. We have found that combinations of low-level models well-chosen to suit a particular task can outperform single

more sophisticated models that do not suit the task well. We have also found cases where a single sophisticated model can outperform combinations of low-level models. What is needed is a system that can learn how to best exploit multiple models and their combinations, freeing the user from this concern.

Our goal has been twofold: to develop a system that (1) can select the best model when one is best, and

figure out how to combine models when that is best, and (2) can *learn* to recognize, remember, and refine best model choices and combinations, by looking both at the data features and at the user interaction, and thereby increase its speed and knowledge with continuous use. The system FourEyes was developed for this two-part goal.

FourEyes not only looks at precomputed features of the data (as does Photobook), but additionally, FourEyes looks at the user's interaction with the data. The user can give the system examples of data in which the user is interested, e.g., by clicking on some buildings and then on the "positive" example button. The user can also give negative examples, providing corrective feedback to the system. The use of the FourEyes user examples is a kind of relevance feedback, a well-known and powerful technique used in the latest text-based retrieval systems. However, FourEyes goes beyond relevance feedback in its abilities to combine models and to learn.

Given a set of positive and negative examples, FourEyes looks at all the models and determines which model or combination of models best describes the positive examples chosen by the user, while satisfying the constraints of the negative examples. FourEyes is able to choose or combine models in interactive time with each set of positive and negative examples, allowing the features used by the system to change with each query.

FourEyes achieves model combination by multiple stages of processing. Instead of combining features in a numerical feature space (say, by concatenating all the model features into one vector and conducting some kind of subsequent feature selection or linear feature combination), FourEyes abandons numerical feature spaces after they have been used for an initial (first-stage) off-line formation of groupings. This is a key step that distinguishes FourEyes from other existing systems that work with multiple models. The groupings in FourEyes act as a new language through which models can interact; all the models can group all the data, either individually or cooperatively. The problem at this point becomes which models best group the data of interest to the user.

The final stage of the model combination involves an on-line learning method. FourEyes can currently use one of several possible methods (e.g., set cover, decision list, or decision tree) to choose which groupings best cover the user's positive examples, cover none of

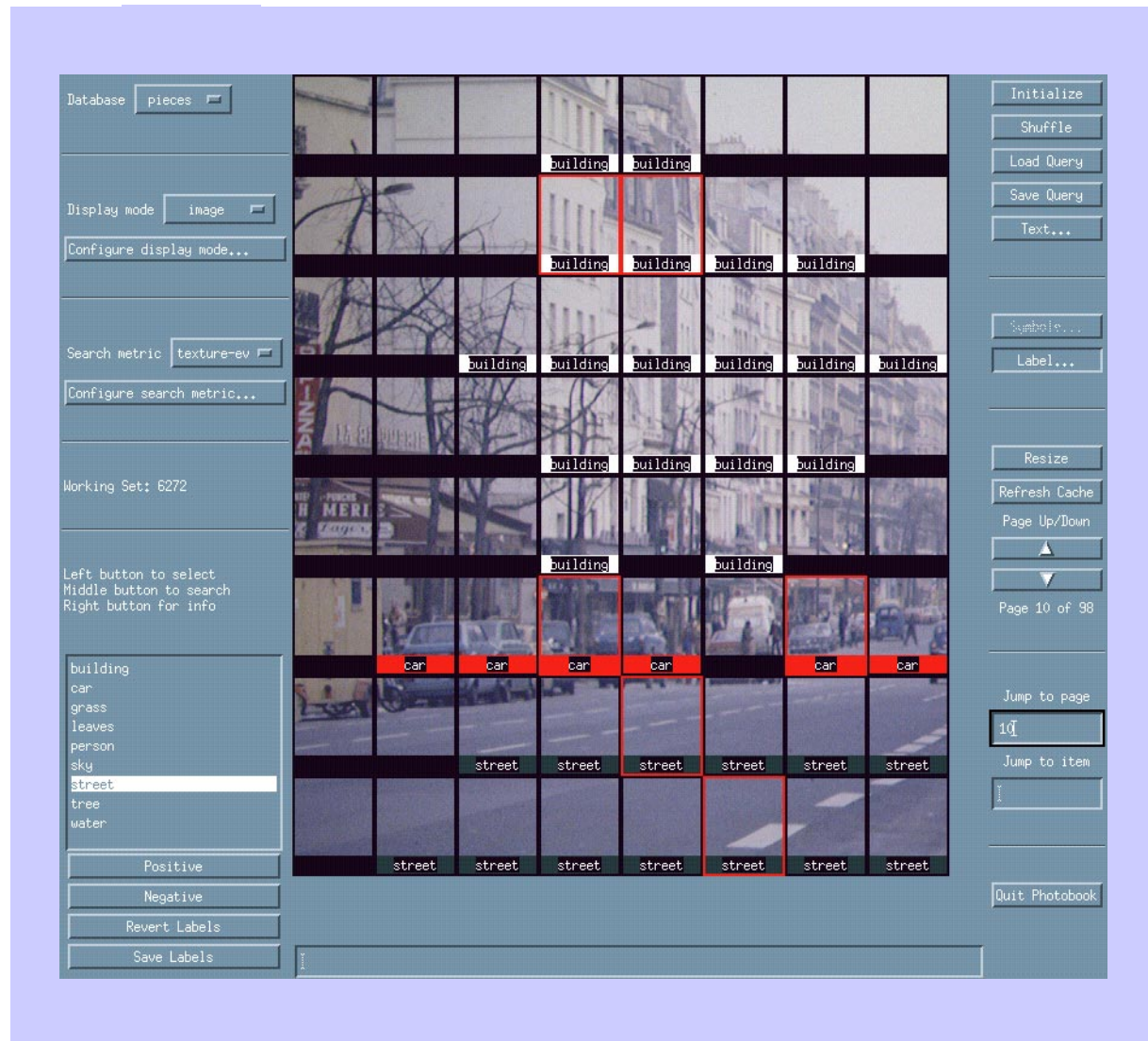
their negative examples, and satisfy some additional criteria. (See References 59 and 60 for details on the learning, as well as on other stages of processing in FourEyes.) The learner can select groupings all from one model, or groupings from any combination of the models available to it. FourEyes might, for example, use a mixture of groupings from motion, color, and texture models.

It is important to emphasize that FourEyes is a learning system; it *learns* which methods of combination best solve a particular problem, and remembers these combinations. In this sense it is quite different from traditional relevance feedback systems. When presented with a new problem similar to one FourEyes has solved before, then FourEyes can solve it more quickly than it could the first time. If the new problem is dissimilar, then FourEyes learns a new combination of models for solving it. FourEyes gets faster as it sees problems similar to those it has seen before. (*Faster* is defined by an ability to retrieve or label the desired concepts given a smaller number of examples of what the user wants.) FourEyes has also demonstrated faster learning across new related (but different) problems.⁶⁰ Current research on FourEyes aims to improve its abilities as a *continuous learner*, using knowledge from problems it has been trained on to improve its performance across new problems for which it has not been trained. This is important in digital libraries, enabling users to change their minds and queries as they see more of the available data.

Power-assisted annotation. Much of image retrieval depends on text descriptions, or *annotations*, which have been tediously typed in by humans. Ideally, semantic annotations and perceptual image features work together, with annotations describing visual relations, and visual features helping to propagate annotations to "visual synonyms."⁶¹ The first version of FourEyes was designed to use vision texture and a society of models to assist the user in annotation.

In annotation, the user labels prototypes in a handful of images, and FourEyes then labels the rest of the database based on the examples of the user. Figure 12 shows an example annotation—the user selected two patches and labeled them "building" (red boxes indicate patches selected by the user), two patches of "car" and two patches of "street." The system then responded by finding the 31 additional labels shown in Figure 12. At the same time, the system went through all the other images in the database, and labeled other places it found to "look like" building,

Figure 12 Screen shot of FourEyes during the labeling of examples of building, car, and street



car, and street. In small-scale tests on a set of vacation photos, this power-assisted annotation process cut the cost of annotating by more than 80 percent.⁶²

Once images are partially annotated, retrieval systems can use semantic search criteria as well as the present visual-feature-based criteria. For example, after using FourEyes to annotate less than 20 percent of the British Telecommunications (BT) image database, queries for “semantically similar” scenes could be made, as illustrated by Figure 13, where an image was retrieved

as similar if it contained a similar percentage of regions with labels of building and street. The location of the labeled regions was not considered, but only their relative area within the image. Effectively the model is a histogram of labels, equipped with a distance on the histogram. At this semantically labeled stage there are many existing tools available that can be used, e.g., an on-line text thesaurus.

It is worth mentioning that no one model available to FourEyes was able to represent the variety of build-

Figure 13 Results after labeling data in FourEyes. “Computer, go find scenes like this one (upper left), with buildings or street.”



ings and streets shown in Figure 13. Instead, FourEyes constructed a concept of “building” and a concept of “street” by combining groupings found by several different models. The exact combinations are transparent to the user, but are learned by the system for speeding up future similar requests.

In general, the performance of power-assisted annotation depends on the data, the nature of the annotations, and the learning algorithm. A benefit of building a learning algorithm into an annotation system is that the FourEyes system saves the most useful label-visual feature associations, essentially constructing a representation that acts as a “visual thesaurus.”⁶¹ A cluster labeled “building” that looks like

white buildings viewed from a sharp perspective can therefore get associated with a cluster labeled “building” that looks like white-trimmed, red brick from a different perspective. Different prototypes of visual building become linked to the same semantic label. Not only does the system accumulate knowledge and improve its performance, but it ultimately helps vision researchers study the connections between high-level visual descriptions and low-level vision texture.

The learning ability of FourEyes allows retrieval algorithms to be customized for each user’s goals, while freeing the user from having to figure out how to hand-set the nonintuitive weights and combinations of the model every time the user’s query goals change.

The reliance on the society of models by FourEyes means that it can simultaneously provide for many notions of similarity—including color, texture, shape, motion, position, and even user-defined subjective associations. The latter are particularly important as many queries (indeed, the most common ones for stock photos in advertising⁶³) are for images with a certain “mood.” Giving computers the ability to learn about affect will make huge new demands on tools for learning and pattern modeling, but is essential for improving their performance in tasks involving human interaction.⁶⁴

Summary

This paper surveys recent research in the Vision Texture group of the MIT Media Laboratory. This research broadens the definition of texture to include all signals best described by collective properties of low-level features—for images, the visual equivalent of “mass nouns.” Several texture models have been investigated, including reaction-diffusion, Markov random fields, cluster-based probability distributions, Wold features, STAR models, and modified particle systems, for describing combinations of visual features that occur in image, video, and graphics. This paper briefly describes each of these models, highlighting their strengths, relations to other models, and potential uses in digital libraries.

Understanding multiple models and their interactions is an essential part of a greater goal, the construction of an effective “society of models.” The society of models approach allows a system to flexibly choose the best solution, whether it is a combination of low-level models or a single sophisticated model. This approach is especially important in interactive systems for image browsing and retrieval, where a variety of models tailored to different goals are necessary for best performance.

This paper describes two such systems for interactive browsing, retrieval, and annotation of image and video data. One of these systems, FourEyes, looks not only at precomputed features of the data (like the other system, Photobook), but also looks at the user’s interaction with the data. Using a learning algorithm, FourEyes determines which models or combinations of models perform best for the user’s task. It accumulates knowledge from the user, becoming more effective with increased use. Together, the vision texture models and learning algorithm contribute to new systems that save the user’s time organizing, manipulating,

browsing, querying, and annotating large sets of visual information.

Additional technical information on this topic is available in Reference 65.

Acknowledgments

It is a pleasure to acknowledge the graduate students who have contributed to the research described in this survey: Monika Gorkani, Fang Liu, Tom Minka, Chris Perry, Kris Popat, Alex Sherstinsky, and Martin Szummer. Marty Friedmann contributed the multiple-models football scenario, and caused me to think about how humans switch among models with different strengths. I would also like to acknowledge the anonymous reviewers for their helpful comments. This work has been generously supported by BT PLC, Hewlett-Packard Co. Research Laboratory, IBM, Interval Research Corporation, and NEC Corporation.

Cited references and notes

1. R. J. Herrnstein, D. H. Loveland, and C. Cable, “Natural Concepts in Pigeons,” *Journal of Experimental Psychology: Animal Behavior Processes* **2**, 285–302 (1976).
2. S. Watanabe, J. Sakamoto, and M. Wakita, “Pigeons’ Discrimination of Paintings by Monet and Picasso,” *Journal of the Experimental Analysis of Behavior* **63** (March 1995).
3. H. D. Hubel and T. N. Wiesel, “Receptive Fields and Functional Architecture of Monkey Striate Cortex,” *Journal of Physiology* **195**, 215–243 (1968).
4. W. T. Freeman and E. H. Adelson, “The Design and Use of Steerable Filters,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-13**, 891–906 (September 1991).
5. R. W. Picard and M. Gorkani, “Finding Perceptually Dominant Orientations in Natural Textures,” *Spatial Vision*, special Julesz birthday issue **8**, No. 2, 221–253 (1994).
6. M. M. Gorkani and R. W. Picard, “Texture Orientation for Sorting Photos at a Glance,” *Proceedings of the International Conference on Pattern Recognition*, Vol. 1, Jerusalem, Israel (October 1994), pp. 459–464.
7. The images from Figures 1, 12, and 13 are part of the BT image database, available by FTP to teleos.com, in VISION-LIST-ARCHIVE/IMAGERY/BT_scenes.
8. M. J. Swain and D. H. Ballard, “Indexing via Color Histograms,” *Image Understanding Workshop*, Pittsburgh, PA (September 1990), pp. 623–630.
9. T. Syeda-Mahmood, “Model-Driven Selection Using Texture,” *Proceedings of the 4th British Machine Vision Conference*, J. Illingworth, Editor, University of Surrey, Guildford, BMVA Press (September 1993), pp. 65–74.
10. R. Polana and R. Nelson, “Low Level Recognition of Human Motion,” *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, Austin, TX (1994).
11. C. B. Thaxton, W. L. Bradley, and R. L. Olsen, *The Mystery of Life’s Origin*, Philosophical Library, New York (1984).
12. C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, Urbana and Chicago, IL (1963).

13. B. B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman and Company, New York (1983).
14. P. S. Stevens, *Patterns in Nature*, Little, Brown and Co., Boston, MA (1974).
15. M. Minsky, *The Society of Mind*, Simon & Schuster, New York (1985).
16. R. Haralick, "Statistical and Structural Approaches to Texture," *Proceedings of the IEEE* **67**, 786–804 (May 1979).
17. M. Tuceryan and A. K. Jain, "Texture Analysis," *The Handbook of Pattern Recognition and Computer Vision*, C. H. Chen, L. F. Pau, and P. S. P. Wang, Editors, World Scientific Publishing Co. (1993), pp. 235–276.
18. A. M. Turing, "The Chemical Basis of Morphogenesis," *Philosophical Transactions of the Royal Society of London Series B—Biological Sciences* **B 237**, 37–72 (1952).
19. A. S. Sherstinsky, *M-Lattice: A System for Signal Synthesis and Processing Based on Reaction-Diffusion*, Sc.D. thesis, Massachusetts Institute of Technology, Cambridge, MA (1994).
20. A. Sherstinsky and R. W. Picard, "M-Lattice: From Morphogenesis to Image Processing," *IEEE Transactions on Image Processing* **5**, No. 7 (July 1996), pp. 1137–1150.
21. R. Kelman, Dow Jones personal communication (1994).
22. A. Sherstinsky and R. W. Picard, "Color Halftoning with M-Lattice," *IEEE Second International Conference on Image Processing*, Washington, DC, Volume II (October 1995), pp. 335–337.
23. C. B. Price, P. Wambacq, and A. Oosterlinck, "Applications of Reaction-Diffusion Equations to Image Processing," *3rd International Conference on Image Processing and Its Application* (1989), pp. 49–53.
24. B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, *Shapes, Shocks, and Deformations I: The Components of Shape and the Reaction-Diffusion Space*, Lab for Engineering Man/Machine Systems LEMS-105, Brown University (June 1992).
25. G. Turk, "Generating Textures on Arbitrary Surfaces Using Reaction-Diffusion," *Computer Graphics* **25**, 289–298 (July 1991).
26. A. Witkin and M. Kass, "Reaction-Diffusion Textures," *Signature* (1991).
27. J. D. Murray, *Mathematical Biology*, Springer-Verlag, New York (1990).
28. M. Hassner and J. Sklansky, "The Use of Markov Random Fields as Models of Texture," *Computer Graphics and Image Processing* **12**, 357–370 (1980).
29. G. R. Cross and A. K. Jain, "Markov Random Field Texture Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-5**, No. 1, 25–39 (1983).
30. R. Chellappa and S. Chatterjee, "Classification of Textures Using Markov Random Field Models," *Proceedings of ICASSP*, San Diego, CA (1984), pp. 32.9.1–32.9.4.
31. R. Chellappa, S. Chatterjee, and R. Bagdazian, "Texture Synthesis and Compression Using Gaussian-Markov Random Field Models," *IEEE Transactions on Systems, Man and Cybernetics* **SMC-15** (March/April 1985).
32. F. S. Cohen, Z. Fan, and M. A. Patel, "Classification of Rotated and Scaled Textured Images Using Gaussian Markov Random Field Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-13**, No. 2, 192–202 (1991).
33. A. I. Mirza, "Spatial Yield Modeling for Semiconductor Wafers," master's thesis, Massachusetts Institute of Technology, Cambridge, MA (May 1995).
34. E. Ben-Jacob and P. Garik, "The Formation of Patterns in Non-Equilibrium Growth," *Nature* **343**, No. 6258, 523–530 (1990).
35. R. W. Picard and A. P. Pentland, "Temperature and Gibbs Image Modeling," MIT Media Laboratory, Perceptual Computing TR #254, Cambridge, MA (1995).
36. I. M. Elfadel and R. W. Picard, "Gibbs Random Fields, Co-Occurrences and Texture Modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-16**, 24–37 (January 1994).
37. R. W. Picard and I. M. Elfadel, "Structure of Aura and Co-Occurrence Matrices for the Gibbs Texture Model," *Journal of Mathematical Imaging and Vision* **2**, 5–25 (1992).
38. R. W. Picard, "Structured Patterns from Random Fields," *Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA (October 1992), pp. 1011–1015.
39. L. Garand and J. A. Weinman, "A Structural-Stochastic Model for the Analysis and Synthesis of Cloud Images," *Journal of Climate and Application Meteorology* **25**, 1052–1068 (1986).
40. R. W. Picard, "Random Field Texture Coding," *Society for Information Display International Symposium Digest*, Boston, MA (May 1992), pp. 685–688.
41. P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, Dover, New York (1966).
42. K. Popat and R. W. Picard, "Novel Cluster-Based Probability Models for Texture Synthesis, Classification, and Compression," *Proceedings of the SPIE Visual Communication and Image Processing* **2094**, 756–768 (November 1993).
43. K. Popat and R. W. Picard, "Cluster-Based Probability Model and Its Application to Image and Texture Processing," to appear; *IEEE Transactions on Image Processing*.
44. N. Saint-Arnaud and K. Popat, "Analysis and Synthesis of Sound Textures," *Proceedings of IJCAI-95 Two-Day Workshop on Computational Auditory Scene Analysis*, Montreal (August 1995), pp. 125–131.
45. N. Saint-Arnaud, "Classification of Sound Textures," master's thesis, Massachusetts Institute of Technology, Cambridge, MA (September 1995).
46. A. R. Rao and J. Lohse, *Identifying High Level Features of Texture Perception*, Computer Science RC17629 #77673, IBM Corporation, Yorktown Heights, NY (1992).
47. J. M. Francos, A. Z. Meiri, and B. Porat, "A Unified Texture Model Based on a 2-D Wold Like Decomposition," *IEEE Transactions on Signal Processing*, 2665–2678 (August 1993).
48. F. Liu and R. W. Picard, "Periodicity, Directionality, and Randomness: Wold Features for Image Modeling and Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to appear; also MIT Media Laboratory, Perceptual Computing TR #320, Cambridge, MA.
49. S. A. Niyogi and E. H. Adelson, "Analyzing and Recognizing Walking Figures in XYT," *CVPR*, Seattle, WA, Computer Vision and Pattern Recognition, IEEE Computer Society Press (June 1994), pp. 469–474.
50. A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, NJ (1989).
51. M. Szummer and R. W. Picard, "Temporal Texture Modeling," *Proceedings of ICIP*, Lausanne (1996), to appear.
52. C. H. Perry and R. W. Picard, "Synthesizing Flames and Their Spreading," *Proceedings of the Fifth Eurographics Workshop on Animation and Simulation*, Oslo, Norway (September 1994).
53. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "The QBIC Project: Querying Images by Content Using Color, Texture, and Shape," *Storage and Retrieval for Image and Video Databases*, W. Niblack, Editor, San Jose, CA, SPIE (February 1993), pp. 173–181.
54. H.-J. Zhang, S. W. Smoliar, J. H. Wu, C. Y. Low, and A. Kanakhalli, *A Video Database System for Digital Libraries*, ISS, National University of Singapore (June 1994).
55. A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Tools for

Content-Based Manipulation of Image Databases," *International Journal of Computer Vision* (1996), in press.

56. "Virage," <http://www.virage.com/> (1995).
57. "IBM QBIC project," <http://www.qbic.almaden.ibm.com/qbic/qbic.html>.
58. R. W. Picard, "Light-Years from Lena: Video and Image Libraries of the Future," *IEEE Second International Conference on Image Processing*, Vol. I, Washington, DC (October 1995), pp. 310–313.
59. T. P. Minka and R. W. Picard, "Interactive Learning Using a 'Society of Models'," to appear in issue of *Pattern Recognition*.
60. T. P. Minka, *An Image Database Browser that Learns from User Interaction*, master's thesis, Massachusetts Institute of Technology, Cambridge, MA (May 1996).
61. R. W. Picard, "Toward a Visual Thesaurus," *Springer-Verlag Workshops in Computing* (1995), to appear; also appears as MIT Media Laboratory, Perceptual Computing TR #358, Cambridge, MA.
62. R. W. Picard and T. P. Minka, "Vision Texture for Annotation," *Journal of Multimedia Systems* 3, 3–14 (1995).
63. D. Romer, *The Kodak Picture Exchange*, seminar at MIT Media Laboratory, Cambridge, MA (April 1995).
64. R. W. Picard, *Affective Computing*, MIT Media Laboratory, Perceptual Computing TR #321, Cambridge, MA (1995).
65. Additional technical notes and computer code are available from our World Wide Web pages, <http://www-white.media.mit.edu/vismod/vismod.html>, and by anonymous FTP from whitechapel.media.mit.edu.

Accepted for publication March 25, 1996.

Rosalind W. Picard, MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: picard@media.mit.edu). Dr. Picard is NEC career development professor of computers and communication, and associate professor of media technology at the MIT Media Laboratory. She earned a B.E.E. from the Georgia Institute of Technology and was named a National Science Foundation fellow in 1984. Dr. Picard earned the M.S. and Sc.D. in electrical engineering and computer science from the Massachusetts Institute of Technology in 1986 and 1991, respectively. She was a member of the technical staff at AT&T Bell Laboratories from 1984 to 1987, and has consulting experience with image and texture modeling, semantic coding, computer vision, pattern analysis, and affective computing. One of the pioneers in algorithms for content-based video and image retrieval, Dr. Picard is currently guest editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* special issue on "Digital Libraries: Representation and Retrieval."

Reprint Order No. G321-5607.