Published in final edited form as:

Pattern Recognit. 2008 March 1; 41(3): 972–982. doi:10.1016/j.patcog.2007.08.007.

Attentive texture similarity as a categorization task: Comparing texture synthesis models

Benjamin Balas^{*}

Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Abstract

Many attempts have been made to characterize latent structures in “texture spaces” defined by attentive similarity judgments. While an optimal description of perceptual texture space remains elusive, we suggest that the similarity judgments gained from these procedures provide a useful standard for relating image statistics to high-level similarity. In the present experiment, we ask subjects to group natural textures into visually similar clusters. We also represent each image using the features employed by three different parametric texture synthesis models. Given the cluster labels for our textures, we use linear discriminant analysis to predict cluster membership. We compare each model’s assignments to human data for both positive and contrast-negated textures, and evaluate relative model performance.

Keywords

Texture similarity; Texture synthesis; Classification; Perception

1. Introduction

While a great deal of work has been done to characterize the nature of “pre-attentive” texture processing, comparatively little research effort has been directed towards the perception of texture under fully attentive viewing conditions. In particular, the question of what image features contribute to perceived similarity between pairs of textures has not been thoroughly investigated. Some attempts have been made to describe latent structures in psychological texture spaces using dimensional models [1–3], non-linear trajectories [4], or cluster-based analysis [5]. In general, studies such as these tend towards the descriptive rather than the quantitative, with the end goal being to characterize the high-level properties that are captured by a particular axis, path, or cluster in the recovered texture space. In other cases, specific image features have been evaluated according to their ability to approximate human judgments of a particular high-level attribute such as “roughness” or “periodicity” [6,7]. While these results are useful for some applications, it is potentially dangerous to extrapolate from such findings to the more general similarity problem as the texture properties isolated for investigation may not be an important aspect of generic similarity judgments.

In the current study, we simultaneously characterize high-level texture similarity psychophysically and computationally. Our basic strategy is to build a dimensional model of texture space by collecting texture groupings from human observers via a card-sorting task. Within this space, we then use *k*-means clustering to determine a set of texture categories. By

^{*}Tel.: +1 617 252 1815. bjbaldas@mit.edu.

performing this second step, we are able to transform the task of judging similarity between pairs of textures into a multi-category classification task. Category membership for each texture is then treated as “ground truth” data that we can attempt to approximate with a simple linear classifier that can operate on any representation of the input images we wish. Currently, we have opted to examine the relative merits of the features employed by three different parametric texture synthesis models. To motivate our use of texture synthesis models as the subject of this study, we briefly discuss alternative texture descriptors used in other domains.

One particular domain that might seem to offer an interesting starting point for our analysis is the modeling of pre-attentive texture segmentation. In segmentation tasks, subjects are usually asked to report the location or orientation of a boundary that separates one region of homogeneous texture from another. Our idea of what features are calculated and compared to accomplish segmentation tasks such as this has evolved over the years to include broad orientation statistics [8], micropatterns within the texture array [9], and the current view that various spatial filters may provide the right outputs to match human performance [10–13]. Unfortunately, the relationship between pre-attentive segmentation and attentive similarity is uneasy at best. Segmentation is generally studied using artificial textures, where it is possible to finely control the extent to which one texture differs from another under some image feature. By contrast, almost all pairs of natural textures would be trivially easy to segment, despite the fact that human observers can usually provide a graded similarity judgment for arbitrary pairs of images. This suggests that the features that prove useful for describing segmentation performance may not be appropriate for characterizing attentive similarity.

An alternative would be to consider features that are used for texture recognition. There have been many independent investigations of the utility of various features for content-based retrieval of textures from complex images, as well as database match-to-query tasks [7,14–16]. One important advantage of such features is that they have generally been developed for use with real images. As such, these features are more capable of capturing the richness of natural textures and may thus be more useful for our purposes. However, it is important to point out that recognition and similarity are not interchangeable, despite being closely related. If one wishes to accurately classify a particular texture in an image, for example, it will be important to use representations that are invariant to changes in scale, illumination, and other sources of image variability. Similarity judgments are not necessarily invariant to these transformations, however. A particular texture viewed from far away may not be judged as highly similar to the same texture viewed close-up, for example. Any recognition algorithm designed to be invariant to a set of transformations on a given texture might have to be re-designed if similarity judgments are highly variable under the same manipulation. To avoid this potential difficulty, we do not consider texture recognition algorithms in the current study (though we cannot rule out their possible efficacy).

In general, the problem of approximating texture similarity judgments is complicated by the lack of any obvious principles governing the mapping between input images and observers’ output. By way of comparison, any rigid 3-D object is limited in its ability to take on different appearances by its geometric and photometric properties. Though the set of appearances such an object can exhibit may lie on a complicated manifold in image space, these appearances are ultimately constrained by a small number of true degrees of freedom such as pose, lighting angles, etc. [17]. Images of texture are far less constrained in this regard, since a particular texture may be recognizable from multiple distinct local image patches with unique geometry. Worse than this, judging similarity (rather than identity) is even more difficult since highly similar textures may share little in terms of true shape or material properties, resembling one another only in appearance. Thus, a collection of similar-looking textures may not necessarily lie on any particular manifold through image space, or be easily describable in terms of a limited number of degrees of freedom.

Given that texture similarity may be fundamentally dependent on appearance (rather than underlying form), we have attempted here to model similarity judgments using the image features that form the basis of three distinct texture synthesis models. Synthesizing a convincing texture is fundamentally dependent on discovering a small set of features that can capture texture appearance effectively. In general, there are no assumptions made about invariances that should be accounted for (as in texture recognition), nor is the texture description meant to account for a particular psychophysical task (such as texture segmentation). Instead, texture synthesis models are designed to measure the “ingredients” of a texture so that arbitrarily large amounts of that same texture can be constructed from noise images. The criterion for success is simply the appearance of the synthesized texture. A wide range of features have been employed in synthesis models, ranging from non-parametric models that use pixel-level [18] or patch-level representations [19], to parametric methods incorporating V1-like filters [20–23]. Aside from being a useful graphics application of texture research, we have recently demonstrated [24] that synthesis models provide a means for understanding what statistics are important for capturing the structure of various natural textures under pre-attentive conditions.

Texture synthesis models are thus a particularly useful starting point for modeling attentive texture similarity. Such models are built to address some of the same issues we encounter in understanding similarity, namely determining the “ingredients” of a texture that contribute most to its appearance. Good models are defined by their ability to accurately capture appearance with a set of candidate features. This gives us an intuitive means of guessing what features we expect to do a good job of approximating the human categorization judgments we collect. To the extent that a model measures the right things to construct a convincing synthetic image, we can expect that it is using features that are perceptually important to our observers. In turn, we would expect that these perceptually important features may provide a good basis for determining whether or not two textures are similar, and should therefore be grouped together. Thus, we would naively expect high-quality synthesis to imply good performance in a similarity task.

In the current study, we evaluated the performance of feature sets used in three distinct parametric models of texture synthesis. First, we used the power spectrum recovered from a texture to approximate its appearance [25]. Second, we used the Heeger–Bergen analysis and synthesis algorithm to characterize textures in terms of oriented contrast energy at multiple spatial scales as measured by derivative-of-gaussian features [20]. Finally, we used the Portilla–Simoncelli algorithm, which augments basic measurements of multi-scale-oriented contrast with higher-order measurements of edge co-occurrence across orientation, position, and scale [21,26,27]. These three algorithms represent a clear progression in synthesis quality over the span of approximately 15 years of research, making it easy to determine whether or not there is a clear relationship between synthesis quality and similarity performance. Also, the algorithms differ a good bit in terms of basic primitives, incorporating features that capture global (power-spectrum), local (Heeger–Bergen), and intermediate image structure (Portilla–Simoncelli). We discuss the details of each model in a later section.

We conducted our analysis on natural texture images. We also assessed agreement between model and experiment on both positive and contrast-negated textures. This manipulation is important from both a cognitive and computational standpoint. Negation often greatly disrupts human performance, despite the fact that many low-level aspects of the image are preserved. For example, faces are particularly difficult to recognize in photographic negative [28]. This impairment has been attributed to the disruption of information vital for computing shape-from-shading [29], the disruption of surface pigmentation [30], or to an impaired ability to extract certain material properties like translucency [31]. Given the interesting perceptual consequences of contrast negation, we expect that subjects’ judgments may change dramatically when images are negated. In particular, since subjects may be less able to identify

the materials or objects that make up the texture, our models may be more able to accurately predict category membership in the negative case than the positive case.

In general, we do not expect to find profound agreement between human and model judgments. However, we feel that this analysis is potentially quite valuable nonetheless. Given the complexity of the problem it is worthwhile to see how much we can accomplish with feature sets that have proven extremely useful in another domain. Also, reformulating texture similarity as a categorization task may prove to be a useful tool for comparing the efficacy of texture representations beyond those discussed here. We continue by presenting the three synthesis algorithms in more detail, discussing the features used in each model to represent target textures.

2. Algorithms for texture synthesis

As we stated above, the goal of texture synthesis algorithms is to describe a procedure for measuring enough statistics in a target image that a new image can be created that will strongly resemble the original texture. We consider three parametric texture synthesis algorithms in this study, each one based on a unique set of features. Here, we describe each algorithm, paying particular attention to the set of measurements used to describe each target texture.

2.1. Power-spectrum texture painting

This algorithm is the simplest one we will consider and represents one of the earliest attempts to create synthetic digital texture. In this procedure, target textures are subjected to a Fourier transform. The phase terms are discarded, and the power spectrum of the target image is used to describe its appearance. A new texture can be synthesized by creating a new phase image (perhaps taken from a Fourier-transformed noise image) and then carrying out an inverse Fourier transform using the target power spectrum and the newly created phase information. The resulting image is the synthetic texture. For our classification experiments, we down-sampled the original 512×512 pixel image by one octave (using a binomial filter) before calculating the power spectrum. This was done due to memory limitations on the classification procedure described in the Methods section.

The power-spectrum algorithm can produce good synthetic images of grainy, stochastic textures. It is not very useful for structured textures containing discrete elements or for non-homogeneous textures.

2.2. The Heeger–Bergen algorithm

The second algorithm we consider is the Heeger–Bergen texture synthesis algorithm. This model measures local oriented contrast at multiple scales of the target image and then attempts to alter a noise image to have the same filter output statistics as the target, while maintaining the original pixel histogram of the target texture.

To be more specific, each target image is filtered at multiple scales with oriented directional derivative spatial filters. The expression for such a filter (a first-order derivative-of-Gaussian oriented at an angle θ) is as follows:

$$f(x, y) = 2x \cos(\theta) e^{-(x^2 + y^2)/\sigma^2} + 2y \sin(\theta) e^{-(x^2 + y^2)/\sigma^2}.$$

Each filtered image (at one orientation and one spatial scale) is then compressed into a histogram. The target image is thus expressed as a set of sub-band histograms, each providing a summary of the filter outputs obtained by filtering the target texture at a particular scale and orientation. Furthermore, an intensity histogram of the target texture is also maintained. To

create a synthetic version of the target texture, a noise image is initialized and then altered to have the same sub-band histograms as the target texture at each scale and orientation. Following this manipulation, the intensity histogram of the target is imposed on the synthetic image to remove any artifacts of the sub-band histogram matching process. These two histogram-matching steps are iterated until the amount of image change at each iteration is below a pre-determined threshold, or until a set number of iterations is complete. For our purposes, we used 17×17 pixel first-order derivative-of-gaussian filters at four orientations (0° , 45° , 135° , 90°), with a spatial constant of 5 pixels. These filters were applied at four spatial scales by down-sampling the target image an octave at a time (again using a binomial filter) and filtering the down-sampled image with the 17×17 pixel oriented filter described above. The pixel intensity histogram and all sub-band histograms had 256 bins each.

The Heeger–Bergen algorithm is useful for creating synthetic images of a much wider range of target textures than the power-spectrum algorithm. However, textures with extended contours or large-scale structures are not well approximated by this technique due to its basic characterization of the image solely in terms of local contrast.

2.3. The Portilla–Simoncelli algorithm

Finally, we present the Portilla–Simoncelli algorithm. This is a very complex model of texture synthesis, a full description of which is beyond the scope of this paper. The interested reader is referred to the original report of this algorithm for implementation details [21]. Here, we briefly summarize the features measured in the analysis of a target texture.

This model functions in much the same way as the Heeger–Bergen algorithm, insofar as a noise image is manipulated to have the same statistics as a target texture. Furthermore, this algorithm is also based on a pyramid decomposition of the target texture into filtered images at multiple scales and orientations. Just as described above, target textures are down-sampled and filtered at multiple orientations using oriented derivative-of-gaussian filters. Unlike the Heeger–Bergen algorithm, however, the filter outputs are not simply listed in a histogram. Instead, the co-occurrence of filter outputs across space, scale, and orientation is measured in multiple ways which we describe below.

The Portilla–Simoncelli model utilizes four large sets of features to generate novel texture images from a specified target. The first of these feature sets is a series of first-order constraints on the pixel intensity distribution derived from the target texture. The mean luminance, variance, kurtosis, and skew of the target are imposed on the new image, as well as the range of the pixel values. The skew and kurtosis of the low-resolution version of the image produced by the pyramid decomposition is also included here. Second, the local autocorrelation of the target image's low-pass counterparts in the pyramid decomposition is measured and matched in the new image. Third, the measured correlation between neighboring filter magnitudes is measured. This set of statistics includes neighbors in space, orientation, and scale. Finally, cross-scale phase statistics are matched between the old and new images. This is a measure of the dominant local relative phase between coefficients within a sub-band, and their neighbors in the immediately larger scale.

Since the Portilla–Simoncelli algorithm includes measurements of feature correlations across scale, orientation, and spatial location, it is capable of successfully capturing extended contours and large-scale inhomogeneities in the target. For our purposes, we use default settings of the original implementation of the model as described in the authors' initial report [21] (a MATLAB implementation of this code is available online at <http://www.cns.nyu.edu/~lcv/texture/>).

We conclude this section with a figure depicting the synthetic texture resulting from applying each of these models to the same target texture (Fig. 1). There are clearly dramatic differences in synthesis quality. In our classification experiment, we determine whether these clear differences in synthesis ability translate into comparable differences in the ability of each feature set to serve as the basis for predicting human judgments of texture similarity.

We continue by describing our method for obtaining clusters of similar-looking textures from human observers.

3. Identifying perceptual texture families using human similarity judgments

We characterized the perceptual similarity of a large group of natural textures via a set of clusters obtained from human observers during a sorting task. These clusters are used to assign labels to the textures comprising each group, which allowed us to evaluate each set of candidate features in terms of classification accuracy in a leave-one-out cross-validation test. Here we describe our method for obtaining these clusters. We carried out our analysis using both unaltered, positive images of natural textures and contrast-negated versions of the same images.

3.1. Methods

3.1.1. Subjects—We recruited 48 subjects to participate in this task from the MIT undergraduate community. Subject age ranged from 18 to 40 years of age, and all subjects reported normal or corrected-to-normal vision. All subjects were naïve to the purpose of the experiment. Twenty four volunteers contributed to the sorting of the positive texture images, and the remaining 24 volunteers contributed to the sorting of the negative texture images. Within each group of 24 participants, 16 people carried out the initial sorting of “training images,” with the remaining eight people carrying out the classification of “test images.” This two-step process made the initial sorting task more tractable for our participants while still providing category data for a large number of textures.

3.1.2. Stimuli—One hundred and twelve textures taken from the Brodatz texture collection [32] were used in this experiment. Thirty of these textures were removed to serve as the “training images” and have been used in previous studies of high-level texture similarity [1] because they represent a convenient cross-section of the full Brodatz collection (Fig. 1). The original 8.5×11 in images were scanned in at a resolution of 72 pixels/inch. For easy handling, these were reduced in size to approximately 2×3 in and printed on white cardstock at 1200 dots per inch. For computational analysis, a 512×512 pixel patch was taken from the center of each original image. Subjects carried out the sorting task on a large table under fluorescent illumination.

The Brodatz images employed in Experiment 1 were contrast-reversed in Matlab. Original pixel values ranged from 0 to 255, and negation was accomplished by subtracting each original value from 255. Images of some of the textures used in the sorting task are displayed in Fig. 2.

3.1.3. Procedure

3.1.3.1. Sorting of “training images”: Sixteen observers in each group (positive and negative textures) were given the stack of 30 “Training” cards and asked to form groups of similar-looking textures. They were told to use any visual criterion that they felt was relevant, but to refrain as much as possible from grouping textures together according to object identity. For example, a picture of tree bark viewed up close should not necessarily be put in the same group as picture of a copse of trees just because both images contain tree parts. Rather, we suggested that subjects put textures into the same group only if they felt there was a true visual similarity

between the images within a group. Subjects were free to form as many groups as they wished, and were free to take as much time as necessary to complete their sorting. Subjects typically completed the task in 20 min or less.

3.1.3.2. Multidimensional scaling and clustering: The groups of textures created by our observers were used to construct two matrices of pooled pair-wise texture similarity ratings, one for the positive textures and one for the negative textures. In each case, a 30×30 matrix of zeros was initialized. Then, for each group of textures formed by a subject, a “1” was placed in the i th row and j th column of the matrix if textures i and j were both present in the group. The binary matrices obtained from all subjects were summed, resulting in a similarity matrix where a high value at entry i, j indicated strong similarity between textures i and j . Each texture was considered to always be in a group with itself, so each entry along the main diagonal of this matrix was a “16” (the maximum possible similarity score across our 16 subjects).

Since we needed to translate similarity ratings into distances, we subtracted each entry in this matrix from 16. Given this dissimilarity matrix, we positioned our 30 textures into a “psychological” similarity space through the use of a classical multi-dimensional scaling (MDS) algorithm [31].

Typically, MDS is used to recover the dimensions along which points are organized. However, we did not make any use of the resulting axes, other than to provide a means for positioning each texture in a space that respects the “distances” calculated across our subjects’ groupings. We used three dimensions to characterize these textures, following the example of Rao and Lohse [1]. The subsequent clustering procedure was not highly sensitive to the number of included dimensions, however, meaning that this parameter does not likely require fine tuning.

We continued by using k -means clustering [33] to discover clumps of textures that are similar to each other, as quantified by their position in the 3-D texture space defined by MDS. The k -means algorithm requires that the user specify a value of k , which is the number of clusters to be discovered in the data. Since, there is no *a priori* way to know how many clusters we should be looking for in either case we first conducted a simple graphical analysis on the scree plots constructed from clustering solutions on both groups of textures using values of k between 2 and 30. Graphs such as these are often used to determine intrinsic model complexity (see Ref. [17] for some interesting examples). Based on these scree plots, a 7-cluster solution was selected for each group (see Figs. 3 and 4).

Why recover clusters rather than attempt to model the dissimilarity matrix itself? While this is possible in principle, we prefer the clustering analysis for several reasons. First, the dissimilarity matrix only represents the pooled data from multiple observers’ categorization judgments. Since individual subjects only provided categories, not paired similarity ratings, we restrict our models to the same judgment. Second, most pairs of texture will be highly dissimilar. Using categories rather than the full dissimilarity matrix allows us to ignore a very large number of texture comparisons that are likely not meaningful to the human observer. Finally, to model the paired similarity of all the textures in our database would require an intractably large number of trials for our observers. By using the similarity space generated by MDS solely for extracting clusters, we are able to rapidly collect category labels for all of our images. We continued by describing how the remaining textures in our database were categorized into the groups defined by training image sorting.

3.1.3.3. Sorting of “test images”: After clustering the training images, our next step was to determine how observers would assign the remaining “test” textures to these categories based on visual similarity. Volunteers that participated in this second portion of our task were presented with the “training images” laid out in their respective clusters. They were then given

the remaining 82 “test images” and asked to place each new texture in front of the cluster they felt it best belonged to. New textures were placed face down to stop subjects from forming a “running average” of the clusters they were augmenting. If participants felt that a given texture could not be reasonably assigned to any cluster, they could discard the image without providing an assignment.

For each “test image” in the positive and negative groups a cluster label was assigned to an image according to the majority of observers’ votes. Images were discarded from further analysis if there was no majority across votes, or if the majority response was “no category.” Two textures were removed from both the positive and negative groups as a result. The full lists of cluster assignments for the positive and negative cases are listed in Tables 1 and 2.

Having obtained category labels for our full set of 110 textures we used these labels as ground truth data for testing the efficacy of the features used in the three texture synthesis models discussed earlier. We continue by describing our classification procedure.

4. Modeling similarity judgments using classification

To compare the utility of each feature set, we used the labeled texture images to carry out multi-class classification. We chose to use relatively simple classifiers for this experiment, both because we have no *a priori* way to assess what methods would be most appropriate for this problem domain and also since we are particularly interested in relative model performance rather than absolute accuracy.

4.1. Dimensionality reduction

All of our candidate feature sets have very high dimensionality and substantial feature redundancy, so we first carried out principal components analysis (PCA) on the feature vectors obtained from each synthesis model. Classification was then carried out given these low-dimensional embeddings of the texture features. This is commonly done in many computer vision tasks, such as face recognition [34]. At test, we also parametrically varied the number of principal components used to represent the data in each model to determine the relationship between accuracy and dimensionality for all models under consideration.

4.2. Classification methods

For each set of features, we measured leave-one-out classification accuracy using discriminant analysis. For each test case, we determined the *a posteriori* probability $P(C_j|x)$ that the test point ‘ x ’ belongs to the cluster C_j for all j clusters. We computed this probability by combining the likelihood term $P(x|C_j)$ with the prior probability $P(C_j)$ using Bayes’ Law.

The likelihood $P(x|C_j)$ was calculated for each test point by fitting a multivariate normal probability density function to each cluster in feature space. Formally, the probability that a point at location x came from the cluster C_j is as follows:

$$P(x|C_j) = \frac{1}{(2\pi)^{N/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^t \Sigma^{-1} (x - \mu)\right),$$

where μ is the centroid of the cluster C_j and Σ is an $N \times N$ covariance estimate for that cluster. We calculated the covariance matrix in three distinct ways, described below:

1. The ‘diagonal linear’ discriminant determined a diagonal estimate of covariance that was pooled across the texture groups.

2. The ‘linear’ discriminant determined a non-diagonal, pooled estimate of covariance for the groups.
3. The ‘diagonal quadratic’ discriminant determined a diagonal estimate of covariance stratified by texture group.

The prior term $P(C_j)$ was set by counting the number of training points belonging to each cluster. Cluster membership of each test point was decided according to a MAP rule, in which the cluster with the highest posterior probability was selected.

4.3. Results

For each synthesis model, we used leave-one-out accuracy as our measure of categorization performance. The earlier distinction between “training” and “test” images is no longer used. Instead, we categorized each texture using the labeled data from the remaining 109 images. Given the complexity of the task, our first question was whether or not any of these models performs better than expected by chance (~ 14% for a 7AFC task). Second, we asked whether any model proved better than the others at predicting human judgments. Our initial hypothesis was that texture synthesis models that are capable of synthesizing a wide range of textures with high quality should perform better than those that are very limited. Thus, we expected the Portilla–Simoncelli model to do best, followed by Heeger–Bergen and the power-spectrum model in that order. We also varied the number of principal components used to represent the data from 1 to 100. This allowed us to see how robust each feature set was to compression. Finally, using the positive and negative texture groups allowed us to investigate whether texture negation results in the formation of texture groups that are more amenable to classification via low-level image statistics. The results of this analysis for both the positive and negative textures are presented in Fig. 5.

First, examining the data from the positive textures we can see that all of our models perform better than chance. It is also apparent that the power-spectrum model does best, followed by the Portilla–Simoncelli model and lastly, the Heeger–Bergen model. In terms of how compression affects categorization, none of our models improve much when more than about 15 principal components are included. In particular, the power-spectrum model reaches its peak performance with a relatively small number of PCs.

Second, examining the performance with negative textures, we see that our hypothesis concerning overall improvement in performance following negation is not supported by these results. In general, we can see that accuracy is not appreciably different between the positive and the negative case. If anything, negative classification performance is a bit lower. While contrast negation substantially altered observers’ groupings, they did not change such that classification with the features we have used here was any easier. Also, there are no clear differences in model performance in this case. Overall, we conclude that negation has profound effects on human similarity judgments, but not on the efficacy of these features to approximate those judgments.

Finally, performance is not substantially different across our three classifiers. Incorporating more detailed measurements of feature covariance is not helpful in these data sets, suggesting that mislabeled data points may generally be quite distant from their ‘parent’ density, or overlap substantially with other clusters.

4.4. Discussion

Human subjects bring a whole host of visual and cognitive processes to bear when assigning textures to clusters based on similarity, and so the level of performance achieved by our models is very encouraging. It is likely that even the most complete model of spatial vision would

prove inadequate for modeling all the intricacies of these judgments, so being able to accurately predict human judgments at this level with such simple methods is remarkable. More importantly, we can see differences in performance between our three candidate models. This indicates that our reformulation of texture similarity as a multi-class categorization task is a useful methodology for comparing different feature sets. In particular, the result that the power-spectrum model is best able to match human judgments in the positive case is surprising. Of the three models proposed, the power-spectrum model is the least capable of successfully reconstructing an arbitrary target texture. Our basic hypothesis that successful synthesis should imply success at estimating similarity is contradicted. The superiority of the power-spectrum model in this experiment demonstrates that better classification performance does not depend on more accurate reconstruction of the image.

Finally, how might we achieve better performance? We could attempt to use a more powerful classifier than the discriminants we employed here. The use of a support-vector machine with a polynomial or Gaussian kernel, for example, might increase the accuracy of all of our models. However, before employing more complex methods for any one model in isolation, it may also prove useful to consider combining information from all three. The simplest way to do this is to combine the measurements made by all three models into one large input matrix. However, there are more sophisticated methods for combining a set of weak classifiers into a far more powerful classifier. This is commonly referred to as “boosting,” and has proven successful for a range of classification tasks, including face recognition [35].

Though the categorization judgments made by these three parametric models might be combined in some way to achieve better performance, it is likely that future efforts will also benefit from considering a wider range of features. For example, finding features that reflect the constraints imposed in non-parametric texture synthesis models may prove extremely useful. Fragment-based representations in particular have proven useful in object recognition [36]. Defining “informative” fragments for a category of similar textures is likely to be quite challenging, but may be very powerful. We hope that by providing the category labels we obtained from our sorting task, other researchers will be encouraged to attempt modeling these data by other means.

5. Conclusions

The current results provide some interesting insights into the perception of similarity in natural textures. First of all, we have demonstrated that texture similarity can be conceived as a multi-class categorization task. This makes it possible to apply many established computer vision and statistical learning techniques to the problem of predicting perceived similarity between pairs of textures. Classification accuracy is a simple quantitative means of judging how well a particular model approximates human judgments. We have further demonstrated that it is possible to make meaningful comparisons between different feature sets using this approach. Second, our results are rather surprising in that the poorest model of texture synthesis provides the best approximation to human data. While the power spectrum of a texture is a relatively coarse measurement of image structure, it provides the most predictive power of any of our models. Finally, we have shown that when images are negated, human similarity judgments are substantially different, though the relative performance of the models we considered was not greatly affected. This may provide an important constraint to guide the search for more effective feature sets.

Acknowledgments

The author would like to thank Ann Breckencamp for all of her assistance in constructing stimuli and collecting similarity judgments for use here. Also, the observations and advice of Ted Adelson, Erin Conwell, Aude Oliva, Ruth

Rosenholtz, Richard Russell, and Pawan Sinha proved invaluable. BJB is supported by a National Defense Science and Engineering Graduate fellowship.

References

1. Rao AR, Lohse GL. Towards a texture naming system: identifying relevant dimensions of texture. *Vision Res* 1996;36:1649–1669. [PubMed: 8759466]
2. Harvey LO, Gervais MJ. Internal representation of visual texture as the basis for the judgment of similarity. *J Exp Psychol Hum Percept Perform* 1981;7:741–753. [PubMed: 6457089]
3. Gurnsey R, Fleet DJ. Texture space. *Vision Res* 2001;41:745–757. [PubMed: 11248263]
4. Richards W, Koenderink JJ. Trajectory mapping (TM): a new non-metric scaling technique. *Perception* 1995;24:1315–1331. [PubMed: 8643335]
5. Heaps C, Handel S. Similarity and features of natural textures. *J Exp Psychol Hum Percept Perform* 1999;25:299–320.
6. Fujii K, Sugi S, Ando Y. Textural properties corresponding to visual perception based on the correlation mechanism in the visual system. *Psychol Res* 2003;67:197–208. [PubMed: 12955509]
7. Liu F, Picard RW. Periodicity, directionality, and randomness: wold features for image modeling and retrieval. *IEEE Trans Pattern Anal Machine Intell* 1996;18:722–733.
8. Beck J. Perceptual grouping produced by changes in orientation and shape. *Science* 1966;154:538–540. [PubMed: 5916950]
9. Bergen JR, Julesz B. Rapid discrimination of visual patterns. *IEEE Trans Syst Man Cybern* 1983;13:857–863.
10. Bergen JR, Adelson EH. Visual texture segmentation based on energy measures. *J Opt Soc Am A* 1986;3:99.
11. Bergen J, Adelson E. Early vision and texture perception. *Nature* 1988;333:363–364. [PubMed: 3374569]
12. Bergen, JR.; Landy, MS. Computational Modeling of visual texture segregation. In: Landy, MS.; Movshon, A., editors. *Computational Models of Visual Perception*. MIT Press; Cambridge, MA: 1991.
13. Malik J, Perona J. Preattentive texture discrimination with early vision mechanisms. *J Opt Soc Am A* 1990;7:923–932. [PubMed: 2338600]
14. De Bonet, JS.; Viola, P. Texture recognition using a non-parametric multi-scale statistical model. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 1998. p. 641–647.
15. Long, H.; Tan, CW.; Leow, WK. Invariant and perceptually consistent texture mapping for content-based image retrieval. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 2001. p. 117–120.
16. Ma, WY.; Manjunath, BS. Texture features and learning similarity. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 1996. p. 425–430.
17. Tenenbaum JB, de Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. *Science* 2000;290:2319–2323. [PubMed: 11125149]
18. Efros, AA.; Leung, TK. Texture synthesis by non-parametric sampling. presented at *Proceedings of the International Conference on Computer Vision*; Corfu. 1999.
19. Efros, AA.; Freeman, WT. Image quilting for texture synthesis and transfer. presented at *SIGGRAPH '01*; Los Angeles, CA. 2001.
20. Heeger, D.; Bergen, J. Pyramid-based texture analysis/synthesis. presented at *ACM SIGGRAPH*; 1995.
21. Simoncelli, E.; Portilla, J. Texture characterization via joint statistics of wavelet coefficient magnitudes. presented at *Fifth IEEE International Conference on Image Processing*; Chicago. 1998.
22. Zhu, SC.; Wu, YN.; Mumford, D. Filters, random fields and maximum entropy (FRAME)—towards the unified theory for texture modeling. presented at *IEEE Conference on Computer Vision and Pattern Recognition*; 1996.
23. Zhu SC, Wu YN, Mumford D. Minimax entropy principle and its application to texture modeling. *Neural Comput* 1997;9:1627–1660.

24. Balas B. Texture synthesis and perception: using computational models to study texture representations in the human visual system. *Vision Res* 2006;46:299–309. [PubMed: 15964047]
25. Lewis, J-P. Texture synthesis for digital painting. presented at International Conference on Computer Graphics and Interactive Techniques; 1984.
26. Portilla, J.; Simoncelli, E. Texture modeling and synthesis using joint statistics of complex wavelet coefficients. presented at IEEE Workshop on Statistical and Computational Theories of Vision; Fort Collins, CO. 1999.
27. Portilla J, Simoncelli E. A parametric texture model based on joint statistics of complex wavelet coefficients. *Int J Comput Vision* 2000;40:49–71.
28. Galper RE. Recognition of faces in photographic negative. *Psychon Sci* 1970;19:207–208.
29. Kemp R, Pike G, White P, Musselman A. Perception and recognition of normal and negative faces: the role of shape-from-shading and pigmentation cues. *Perception* 1996;25:37–52. [PubMed: 8861169]
30. Vuong QC, Peissig JJ, Harrison MC, Tarr MJ. The role of surface pigmentation for recognition revealed by contrast reversal in faces and Greebles. *Vision Res* 2005;45:1213–1223. [PubMed: 15733955]
31. Schiffman, SM.; Reynolds, L.; Young, FW. Introduction to Multidimensional Scaling. Academic Press; New York: 1981.
32. Brodatz, P. Textures: A Photographic Album for Artists and Designers. Dover; New York: 1996.
33. MacQueen, JB. Some methods for classification and analysis of multivariate observations. presented at 5th Berkely Symposium on Mathematical Statistics and Probability; Berkeley, CA. 1967.
34. Beveridge, JR.; She, K.; Draper, BA.; Givens, GH. A nonparametric statistical comparison of principal component and linear discriminant subspaces for face recognition. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2001. p. 535
35. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. presented at Accepted Conference on Computer Vision and Pattern Recognition; 2001.
36. Ullman S, Vidal-Naquet M, Sali E. Visual features of intermediate complexity and their use in classification. *Nat Neurosci* 2002;5:682–687. [PubMed: 12055634]

Biography

BENJAMIN BALAS is currently is Ph.D. student in MIT's Department of Brain and Cognitive Sciences, working in collaboration with Pawan Sinha on psychophysical and computational studies of high-level recognition. His recent work focuses on finding novel features for representing objects and textures for recognition, with particular interest in dynamic stimuli.

Fig. 1.

Examples of synthetic textures obtained from the texture synthesis models under consideration in this experiment. The original target texture is pictured at left. There is a striking increase in quality across the different models.

Fig. 2.

The 30 “training” images drawn from the Brodatz database. Subjects placed these images into clusters according to visual similarity in an unconstrained sorting task.

Fig. 3.
 (Lower right) A Scree plot of encoding error vs. number of clusters for positive textures. A seven-cluster solution was selected on the basis of this plot, and the texture clusters thus selected are also pictured with brief high-level descriptors of each group (left).

Fig. 4.

(Lower right) A Scree plot of encoding error vs. number of clusters for negative textures. A seven-cluster solution was selected on the basis of this plot, and the texture clusters thus selected are also pictured with brief high-level descriptors of each group (left).

Fig. 5.

Plots of leave-one-out accuracy as a function of the number of principal components used to summarize the features contained in our three models. The left column displays the results obtained from classification of the positive textures and the right column shows the results obtained from negative textures. Our three discriminant functions (described in the text) are shown by rows with the 'diagonal linear' results at the top, 'linear' results in the middle, and 'diagonal quadratic' results at the bottom.

Table 1

Positive texture families. Each number corresponds to the Brodatz numbering scheme

Positive textures	Cluster number
{10, 16, 22, 35, 36, 37, 50, 51, 68, 76, 78, 79, 80, 81, 85, 87, 105, 106, 110}	Cluster 1
{1, 3, 6, 8, 11, 17, 18, 18, 19, 20, 21, 25, 26, 34, 46, 47, 49, 52, 53, 55, 56, 64, 65, 82, 83, 94, 95, 96, 101, 102, 103, 104}	Cluster 2
{43, 44, 45, 69, 70, 71, 72, 93, 97}	Cluster 3
{5, 23, 27, 28, 30, 31, 48, 54, 66, 67, 74, 75, 98, 111, 112}	Cluster 4
{39, 40, 41, 42, 88, 89, 90, 91, 107, 108, 109}	Cluster 5
{2, 7, 12, 13, 58, 59, 60, 61, 62, 63, 73, 86, 99, 100}	Cluster 6
{4, 9, 24, 29, 32, 33, 38, 57, 77, 84, 92}	Cluster 7

Table 2

Negative texture families. Each number corresponds to the Brodatz numbering scheme

Positive textures	Cluster number
{37, 50, 51, 68, 69, 70, 71, 72, 76, 97, 105, 106}	Cluster 1
{3, 6, 9, 11, 16, 21, 22, 24, 32, 33, 34, 35, 36, 38, 49, 53, 77, 78, 79, 81, 85, 87, 93, 102, 103}	Cluster 2
{2, 5, 10, 12, 28, 30, 31, 54, 111, 112}	Cluster 3
{4, 17, 19, 29, 52, 55, 57, 84, 92, 101, 104}	Cluster 4
{7, 13, 39, 40, 41, 42, 43, 44, 45, 46, 48, 58, 60, 61, 63, 67, 73, 86, 88, 89, 90, 91, 100, 107, 108, 109, 110}	Cluster 5
{1, 8, 18, 20, 25, 26, 47, 56, 64, 65, 80, 82, 83, 94, 95, 96}	Cluster 6
{23, 27, 59, 62, 66, 74, 75, 98, 99}	Cluster 7