A NEW SUBJECTIVE PROCEDURE FOR EVALUATION AND DEVELOPMENT OF TEXTURE SIMILARITY METRICS

Jana Zujovic [a], *Thrasyvoulos N. Pappas* [a], *David L. Neuhoff* [b], *Rene van Egmond* [c], *Huib de Ridder* [c]

[a] EECS Department, Northwestern University, Evanston, IL 60208, USA
{jana.zujovic,pappas}@eecs.northwestern.edu

[b] EECS Department, University of Michigan, Ann Arbor, MI 48109, USA
neuhoff@umich.edu

[c] Faculty of Industrial Design Engineering, Delft Univ. of Technology, 2628 CE Delft, The Netherlands
{r.vanegmond,h.deridder}@tudelft.nl

ABSTRACT

In order to facilitate the development of objective texture similarity metrics and to evaluate their performance, one needs a large texture database accurately labeled with perceived similarities between images. We propose ViSiProG, a new Visual Similarity by Progressive Grouping procedure for conducting subjective experiments that organizes a texture database into clusters of visually similar images. The grouping is based on visual blending, and greatly simplifies pairwise labeling. ViSiProG collects subjective data in an efficient and effective manner, so that a relatively large database of textures can be accommodated. Experimental results and comparisons with structural texture similarity metrics demonstrate both the effectiveness of the proposed subjective testing procedure and the performance of the metrics.

Index Terms— structural similarity metrics, image quality, content-based retrieval.

1. INTRODUCTION

Objective texture similarity metrics are important for a variety of applications, including image and video compression, computer vision, and content-based retrieval. Unlike traditional image quality metrics [1] that evaluate the similarity of two images on a point-by-point basis, texture similarity metrics must allow substantial point-by-point deviations between textures that according to human judgment are quite similar or even essentially identical. The development of such metrics requires extensive subjective tests to fine-tune the metrics and to ensure that their performance agrees with human judgment. However, each application imposes its own requirements on metric performance. For example, in image compression it is important to provide a monotonic relationship between measured and perceived distortion, while in image retrieval applications it may be sufficient to distinguish between similar and dissimilar images, while the ordering within each group may not be important. The application also determines whether an absolute or relative similarity scale is needed. The focus of this paper is on content-based image retrieval (CBIR) but the proposed techniques will also have a significant impact on other applications, such as image compression.

A key challenge in designing subjective tests for metric development and evaluation is the collection of extensive amounts of data from a large database of images in order to capture the essential properties of the problem. Another key challenge, closely linked to experimental design, is the analysis of the recorded data. There exists a rich psychophysical literature on testing procedures and tools for the analysis of the recorded data [2]. However, the well-known and readily-available solutions for test design often have to be substantially modified to fit the constraints and specific needs of a particular application. One of the main problems is striking a balance between the length of the test and fatigue of the subject, and the amount (and quality) of data to be collected, so it can be properly analyzed to extract the desirable information. In addition, if individual preferences are to be addressed, each subject should provide enough data for reliable estimation of the relevant parameters. The success of a subjective test also depends, of course, on providing a set of unambiguous instructions that do not result in any bias.

Depending on the performance requirements, a number of traditional statistical measures can be used in conjunction with subjective tests for metric evaluation. For example, the Spearman's rank correlation coefficient and Kendall's tau rank correlation coefficient can be used when a relative similarity scale is needed [3], while linear regression can be used when an absolute scale is necessary. In both cases, however, a large number of subjective tests is needed in order to compare the objective and subjective similarity between texture

This work was supported in part by the U.S. Department of Energy National Nuclear Security Administration (NNSA) under Grant No. DE-NA0000431. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily relfect the views of NNSA.

978-1-4577-1286-9/11/$26.00 ©2011 IEEE

pairs. The number of comparisons grows quadratically with the number of textures in the database. However, a better understanding of the metric performance requirements can go a long way in reducing the amount of required testing.

For example, in [4], the focus was on the recovery of textures that are "identical" to the query texture, in the sense that they are pieces of the same texture. All that is needed in this case is to start with a database consisting of perceptually uniform textures, which we can then cut into (perhaps partially overlapping) pieces, in order to obtain the test database. Any two pieces that come from the same original (perceptually uniform) texture are then considered identical textures. Thus, the ground truth is known and no further subjective tests are required. In the information retrieval community this known as the *known-item search* [5]. Common measures for this type of retrieval systems include *precision at one* (measures in how many cases the first retrieved document is relevant), *mean reciprocal rank* (measures how far away from the first retrieved document is the first relevant one), *mean average precision* and *precision-recall* plots [4].

In [6], the performance criterion was whether a metric can distinguish between similar and dissimilar pairs, irrespective of the ordering within each group. In this case, the greater the gap in metric values between similar and dissimilar pairs, the better the metric performance. This is the focus of the current paper. To test metric performance in the context of this criterion, we need to organize the test images into clusters of similar textures, preferably with minimal overlap. To accomplish this, we propose a new procedure, Visual Similarity by Progressive Grouping (ViSiProG), whereby subjects are sequentially presented with textures and asked to form groups, which are progressively refined to converge into clusters of visually similar textures. The grouping criterion is that textures blend visually, as if they came from the same tapestry. Semantic grouping is strongly discouraged.

Finally, as we mentioned above, in lossy image compression applications a monotonic relationship between measured and perceived distortion is needed. However, such a relationship makes sense only for similar textures. It is difficult even for humans to quantify the similarity of textures that are not similar [3]. Does it make sense to say that sand and tree bark are more dissimilar than marble and snake skin? Thus, any subjective tests must be limited to similar groups. The proposed ViSiProG procedure is a key preliminary step in conducting subjective tests for compression applications. The overall picture is summarized in Fig. 1, which demonstrates the desired metric performance. Subjective similarity scores are on the horizontal axis and metric similarity values are on the vertical axis. A good metric will result in a monotonic relationship between the two variables in the similar range, and will have a large gap between similar and dissimilar textures.

The similarity of two textures depends on both the color composition and the spatial texture characteristics. However, our previous work indicates [3, 4] that the two attributes should be considered separately. Thus, in the present study we focus only on grayscale textures. However, a similar subjective testing procedure can be implemented for color textures, whether one looks only at composition or at the overall texture.

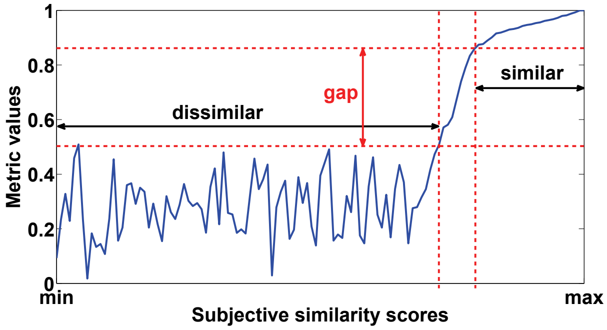


Fig. 1. Ideal performance of texture similarity metric

Our experimental results demonstrate that the proposed procedure collects subjective data in an efficient and effective manner, so that a relatively large database of textures can be accommodated. Comparisons with structural texture similarity metrics are then used to evaluate metric performance.

The remainder of this paper is organized as follows. Section 2 provides a brief overview of texture similarity metrics. The new subjective testing procedure is described in in Section 3 and the experimental setup and results in Section 4.

2. REVIEW OF STRUCTURAL TEXTURE SIMILARITY METRICS

In this section we review the *structural texture similarity metrics (STSIM)* proposed in [6, 3]. These metrics compute the similarity of two images $\mathbf{x}$ and $\mathbf{y}$ by multiplicatively combining a number of terms that compare subband statistics of the two images. The terms to compare can either be computed over the entire image (global window), or over a small sliding window and spatially averaged for an overall metric value. The images are decomposed into subbands using a steerable pyramid. For each orientation and scale we get a similarity score, which we average for the final value for the metric.

For the k-th subband, the *luminance* comparison term is defined as:

$$l^k(\mathbf{x},\mathbf{y}) = \frac{2\mu_x^k\mu_y^k + C_1}{(\mu_x^k)^2 + (\mu_y^k)^2 + C_1} \tag{1}$$

where μ_x^k and μ_y^k are the means of the two windows (local or global). The *contrast* comparison term is defined as:

$$c^k(\mathbf{x},\mathbf{y}) = \frac{2\sigma_x^k\sigma_y^k + C_2}{(\sigma_x^k)^2 + (\sigma_y^k)^2 + C_2} \tag{2}$$

where $(\sigma_x^k)^2$ and $(\sigma_y^k)^2$ are the variances of the two windows. C_1, C_2 are small constants, preventing $0/0$ division. The *first order correlation* terms compare the first order correlation coefficients (autocovariance normalized by the variance) in the horizontal $\rho_x^k(0,1)$ and vertical $\rho_x^k(1,0)$ directions as follows:

$$c_{0,1}^k(\mathbf{x},\mathbf{y}) = 1 - 0.5\left(|\rho_x^k(0,1) - \rho_y^k(0,1)|\right) \qquad (3)$$

The vertical term is defined similarly. These terms are then combined to give a composite measure of structural texture similarity (STSIM) [6] for the k^{th} subband:

$$Q_{\text{stsim}}^k(\mathbf{x},\mathbf{y}) = l^k(\mathbf{x},\mathbf{y})^{\frac{1}{4}} c^k(\mathbf{x},\mathbf{y})^{\frac{1}{4}} c_{0,1}^k(\mathbf{x},\mathbf{y})^{\frac{1}{4}} c_{1,0}^k(\mathbf{x},\mathbf{y})^{\frac{1}{4}} \qquad (4)$$

The STSIM-2 metric proposed in [3] extends the ideas of [6] by including a broader set of local image statistics, motivated by the analysis/synthesis literature [7]. In addition to the terms in (4), STSIM-2 uses terms that compare the cross-correlation between subbands. The luminance, contrast and autocorrelation terms in (1), (2), and (3) are calculated on the *raw* subband coefficients, while the cross-subband correlation statistics are computed on the *magnitudes.* For each orientation, STSIM-2 computes the cross-correlations between subbands at adjacent scales, and for each scale, it computes the cross-correlations between all orientations.

The cross-correlations between the coefficient magnitudes at subbands k and l are normalized by the variances of the two subbands to obtain the cross-subband correlation coefficient

$$\rho_x^{k,l}(0,0) = \frac{E\{(|x_{k,i,j}| - \mu_{x_k})(|x_{l,i,j}| - \mu_{x_l})\}}{\sigma_{x_k}\sigma_{x_l}} \qquad (5)$$

where $|x_{k,i,j}|$ and $|x_{l,i,j}|$ are the magnitudes of the coefficients of subbands k and l, respectively, and μ_{x_k} and μ_{x_l} are the corresponding means of the magnitudes in the window. These are compared as in (3) to obtain a statistic that describes the similarity between the cross-correlations:

$$c_{0,0}^{k,l}(\mathbf{x},\mathbf{y}) = 1 - 0.5\left(|\rho_x^{k,l}(0,0) - \rho_y^{k,l}(0,0)|\right) \qquad (6)$$

Note that the $c_{0,0}^{k,l}(\mathbf{x},\mathbf{y})$ values are in the interval $[0,1]$, just like the STSIM terms.

If the steerable pyramid decomposition yields a total of N subbands, we compute N STSIM maps as in (4), and also M maps (M being a function of the number of scales and orientations in the pyramid) of cross-subband statistics, based on (6). The $N_t = N + M$ matrices are then be combined additively

$$Q_t(\mathbf{x},\mathbf{y}) = \frac{1}{N_t}\left(\sum_k Q_{\text{stsim}}^k(\mathbf{x},\mathbf{y}) + \sum_{k,l} c_{0,0}^{k,l}(\mathbf{x},\mathbf{y})\right) \qquad (7)$$

to obtain a single similarity score.

3. TEXTURE SIMILARITY SUBJECTIVE EXPERIMENT

Gathering subjective scores on texture similarity can be carried out in the conventional way, by asking subjects to rate each and every pair of images and averaging the pooled scores, as was done in [3]. However, there are two serious problems with such conventional approaches. First, as was found in [3], when two texture images are not similar, it is difficult for human subjects to quantify the difference in relative or absolute terms, and there are large inconsistencies between users. There is a similar difficulty in quantifying color differences, for example, whether yellow is more similar to blue than orange is to green. A second problem is that the number of pairs grows quadratically with the total number of images, thus limiting the size of the database on which we can test.

As we argued in the introduction, these problems can be avoided by restricting the range of comparisons to pairs of similar textures (the right most part of Fig. 1). To accomplish this we can divide the experiment into two stages. The first stage consists of forming *similarity clusters*, whereby, textures are similar to each other within a cluster and dissimilar across different clusters. The second stage consists of standard *pairwise comparisons* of images that belong to the same cluster. Thus, by restricting pairwise comparisons to similar textures, this procedure alleviates the problem of quantifying similarity between dissimilar textures, and also, greatly reduces the number of comparisons.

The focus of this paper is on the first stage of the testing procedure, forming similarity clusters of texture images. These clusters are in some sense analogous to *MacAdam ellipses* [8] in color, where each ellipse encompasses the colors that are indistinguishable by human observers from the color at the center of the ellipse. Here, we are working in texture space, and we wish to find the N-dimensional ellipses that contain textures that are considered to be similar by human observers, a relaxed condition compared to MacAdam's perceptually indistinguishable colors. Of course, the criterion could be adjusted to very similar or perceptually indistinguishable textures but the challenge is to go beyond the threshold of detection. This is not a trivial task because textures may differ along several perceptual dimensions, such as contrast, scale, directionality, regularity, periodicity, size and shape of texture elements (textons), average gray level, etc. Preliminary experiments showed that if you ask subjects to group textures into similarity groups, they tend to pick one or more specific dimensions and to ignore other dimensions. Another problem is that subjects may use semantic criteria for the grouping. To avoid such problems, and to encourage subjects to form groups that contain images that are similar in multiple aspects, we tailored our experimental interface so that it relies on visual blending as the similarity criterion.

3.1. Visual similarity by progressive grouping (ViSiProG)

We now present a new approach for forming similarity clusters when given a relatively large set of textures. A key problem in forming clusters when the image set is large, is that it is difficult for a subject to see all the textures on a single computer screen in order to form similarity groups. One solution is to print the textures on paper, and to ask the subject to form groups on a table. However, even then, it is difficult to see and compare all the textures. The main ideas are (a) to build the similarity groups *one* at a time, and (b) to build each group in a step-by-step fashion, picking similar images out of a small subset of images, then repeating the process with another subset of images, until all the images have been considered. Each subject can build several groups in this fashion (one at a time), and the results of multiple subjects can be combined to obtain all the similarity groups in the database. We call the new testing procedure *Visual Similarity by Progressive Grouping (ViSiProG).*

We now describe ViSiProG in more detail. Let the total number of images in the database be N. A set of N_d randomly selected images is displayed on the screen, and the subject is asked to form a group of N_g "most similar" images. For example, in our experiments we used $N = 246$, $N_d = 32$, and $N_g = 9$. Next, the $N_d - N_g$ images that are not included in the group are replaced with a new batch of randomly selected images from the database; the images (N_g from the group and $N_d - N_g$ from the new batch) are shuffled; and the subject is asked again to form a group. This procedure is repeated until the subject has seen all the images in the database. Figure 2 shows a snapshot of the test. The group is formed at the upper left corner of the screen, and is highlighted by a different background color (green in this case). At each iteration, the subject is asked to select N_g images to form a new group out of the N_d displayed images. In the initial stages, the new group does not have to be similar to the group of the previous stage; all the subject is asked to do is form the most similar group. However, as the test progresses and the subject converges on one group of images, then the group is kept together, and the subject is asked to refine it by replacing some of the textures with textures from the new batch. The convergence criterion is the amount of overlap between the groups of two consecutive iterations; in our experiments we set the threshold at 50%, that is, at least five out of nine images must stay the same. If the threshold is met, then the group is kept together for the next iteration; otherwise, the group is shuffled with the new batch of images. This feature allows drifting in the early stages of the test and facilitates convergence in the later stages. The subject can keep refining the group for as long as she/he desires. When the subject has seen all the images in the database at least once, then she/he is given the option of terminating the test and saving the results (the "That's it! Save & close" button appears).

In the first iteration, N_d images are selected randomly with equal probability from the set of N images. In subsequent iterations, each new batch of images is selected randomly from the set of $N - N_g$ images, but the probability of selection decreases with the number of times it has already appeared (and been rejected). This ensures that the subject will see all N images in a relatively small number of trials, but also allows images to be presented multiple times, until the subject converges to a cluster.

The subject forms the group by toggling the check mark at the bottom of each texture. A selection counter makes sure that the subject selects nine textures each time. The subject is allowed to try (by clicking on "show group!") as many texture combinations as desired before proceeding to the next iteration (by clicking on the "Keep my group & shuffle the rest" button). The subject also has the option of rotating each texture (by 90^o at a time) to get a better visual match.

The similarity criterion is one of the keys to the success to achieving our goal of forming texture clusters that are similar across several dimensions. We ask the subject to form a group of textures that blend visually, as if they came from the same tapestry. As we pointed out, semantic grouping is strongly discouraged. As shown in Fig. 2, the group is highlighted by a different background color. Note, that including a border between images facilitates (rather than inhibits) blending, as it masks the discontinuities at the edges of the different textures. A similar effect was observed in tiled displays [9].

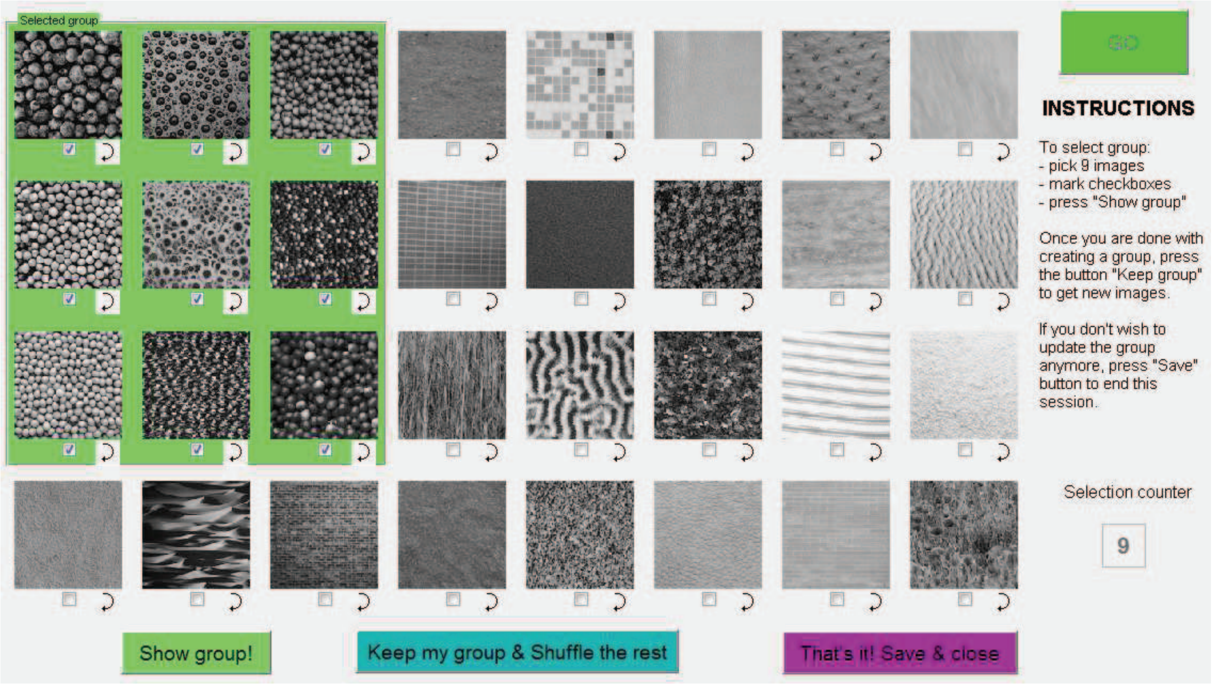
Since there are multiple similarity clusters in the database, the choice of the initial subset of images shown to a subject will greatly influence the resulting cluster. Thus, given random selection of textures, it is very likely that subjects will form different clusters across trials.

4. EXPERIMENTAL RESULTS

In our experiment we used 246 grayscale texture images, out of which 242 were taken from the Corbis database [10] and four are from the authors' personal collection. The images were originally 128×128 pixels, but were downsampled to 100×100 to be presented on the screen. No sampling conversion artifacts were apparent at any stage of the test. We used Matlab GUI development tools, and the users were able to perform the test on their own machines. A total of 15 users participated in this study. They all had normal or corrected-to-normal eyesight. The subjects were asked to execute the test a few times, and the instructions stated that they should neither focus on regenerating the previous group, nor *not* regenerating it, but to try to do the test as if it were the first time they were seeing it. Each user performed the test three times and generated three distinct groups, for a total of 45 groups.

The experimental results were summarized in a similarity matrix S, the off-diagonal entries (i, j) of which indicate how many times image i was in the same group as image j; the diagonal elements, $S(i, i)$ represent the total number of times image i was selected in a group. Out of 246 images, only 134 were selected by the users in a group. The remaining 112 images were never selected. In addition, the rotations of images are taken into consideration, and each rotated image (by $\pm 90°$ or $180°$) was treated as a different image. When this was factored in, the total number of (distinct) images the users selected was 138. Therefore, we can reduce the 246×246 similarity matrix S to form a smaller 138×138 matrix S_{red}.

To analyze the results of the subjective experiment, we used the spectral clustering technique [11]. Spectral clustering methods operate on similarity graphs, which are represented by adjacency matrices, and use the eigenvectors and eigenvalues to cluster the points. These similarity graphs can be formed in different ways, depending on the application. In



INSTRUCTIONS

To select group:
- pick 9 images
- mark checkboxes
- press "Show group"

Once you are done with creating a group, press the button "Keep group" to get new images.

If you don't wish to update the group anymore, press "Save" button to end this session.

Selection counter

9

Fig. 2. Test snapshot

our case, the nodes of the graph represent the images, and the edges and their weights represent the similarity between two images. The adjacency matrix W is formed by scaling the elements of S_{red} by a constant $C = \max(S_{\text{red}})+1$, and placing ones on the diagonal (images are self-similar).

A graph is fully connected if there is an edge between each pair of images. A graph can be reduced by eliminating edges with weights less than some ε, or by keeping the m strongest edges that come out of each node. We chose to use a fully connected graph because we do not want to discard any data gathered from the subjects.

Once the W matrix is generated, the degree of a vertex is defined as:

$$d_i = \sum_{j=1}^{R} W(i,j)$$

where $R = 138$. If we define the *degree matrix* D as a diagonal matrix with entries $d_1, ..., d_R$, the unnormalized graph Laplacian matrix L is calculated as:

$$L = D - W.$$

We refer the readers to [11] for further reading on graph Laplacian matrices. We then compute the R eigenvectors $\{\mathbf{u_1}, ..., \mathbf{u_R}\}$ and eigenvalues $\{\lambda_1, ..., \lambda_R\}$ of L. The spectral clustering algorithm consists of applying the K-means algorithm to those eigenvectors. To cluster into K groups, we use the first K eigenvectors, where each image i is assigned a point p_i in the k-dimensional space with $p_i = (\mathbf{u_1}(i), ..., \mathbf{u_K}(i))$. The number of zero eigenvalues denotes the number of disconnected (non-overlapping) clusters, which in our case was five. By performing K-means clustering on $\{\mathbf{u_1}, ..., \mathbf{u_5}\}$, we get the five non-overlapping clusters the subjects formed.

After forming the initial five clusters, we can partition the graph Laplacian matrix L into five smaller matrices $L_1, ..., L_5$, by grouping the rows and columns that belong to the same clusters. Further analysis of those five clusters is carried out by performing eigenvector decompositions on the smaller Laplacian matrices, and using them as the input to the K-means algorithm.

Since the 5 clusters are connected, there is no strict rule on the number of subclusters into which they should be divided. For our purposes, the most intuitive way to proceed is to look at each cluster in (heavily reduced) texture space defined by the dominant eigenvectors, by placing the images in 2- or 3-dimensional grids. The eigenvectors define for each image a point in space. An example of this visualization is given in Fig. 3, where one of the five clusters is shown. Note that, for the purposes of clarity of the image, not all of its members are present on the plot. Also, note that some of the presented images were selected by users only once, but as mentioned earlier, we did not prune the collected subjective data.

Three of the five clusters had a relatively small number of members (9, 15 and 16) and were not further subdivided. The remaining two clusters had 43 and 55 members. We chose to subdivide them into 3 subclusters each. The choice of the number of subclusters was based on the fact that we wanted stable subclusters, i.e, consistent results of the K-means procedure; at the same time, the subclusters should not be too small (in the extreme case, each image would be a cluster on

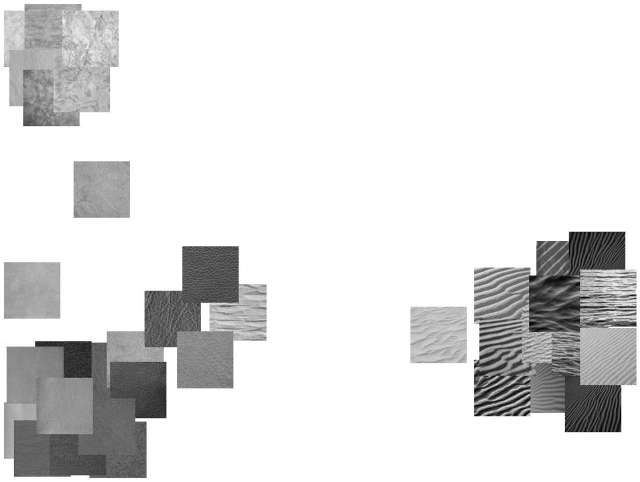


Fig. 3. One of the clusters (with three subclusters)

its own, which would be 100% stable). Setting $K = 3$ satisfied these conditions for both clusters. In $10,000$ K-means runs, one cluster (with 55 members, depicted in Fig. 3) converged to the same partitioning in 96% of cases, and the other one (with 43 members) in 76%. Setting K larger than 3 in both cases resulted in inconsistent partitions and/or merging of the subclusters, thus coming back to the 3 subclusters.

As a result, the 138 images were divided into a total of 9 clusters, counting $2 \times 8, 9, 11, 14, 15, 16, 24$ and 33 members.

4.1. Analysis of STSIM-2 performance

In this subsection, we will compare the performance of the STSIM-2 metric with respect to the clusters obtained by the subjective experiments. One possible solution is to perform spectral clustering on the STSIM-2 similarity matrix, and to try to recreate the nine clusters we obtained by clustering the subjective data. While the adjacency matrix formed by the subjects was very sparse, STSIM-2 assigned a non-zero value to every pair of textures. In that case, spectral clustering is not very effective at sharply separating the clusters.

Another approach is to compare the values of the metric within clusters (intra-cluster similarity values), and across clusters (inter-cluster similarity). This is depicted in Fig. 4. The intra-cluster histogram was computed by pooling all the STSIM-2 values of pairs of images that belong to the same clusters (i.e, STSIM-2 values of all possible pairs within each of the nine clusters). The inter-cluster histogram was computed by pooling all the STSIM-2 values of pairs of images that belong to two different clusters. Both histograms were converted to relative frequencies, since the numbers of possible pairs in intra- and inter-category are different.

As can be seen from the plot, the separation of the STSIM-2 values between similar images (i.e, within one cluster) and dissimilar images (i.e, across different clusters) is not perfect, and there is an overlap in the similarity scores. However, we can also notice that the intra-cluster similarity values are more tightly concentrated around their mean value ($\mu_{intra} = 0.91$), while the inter-cluster values are, in comparison, widely spread ($\mu_{inter} = 0.87$). One of the possible explanations is that we need more subjective experiments to make strong conclusions about which images are considered to be similar, and even more importantly, which images are considered to be dissimilar.

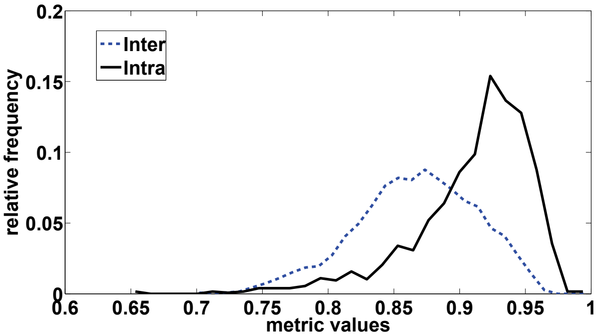


Fig. 4. Histograms of intra- and inter-cluster metric values

5. REFERENCES

[1] T.N. Pappas, *et al.,* “Perceptual criteria for image quality evaluation,” *Handbook of Image and Video Processing*, A.C. Bovik, Ed., pp. 939–959. Academic Press, 2005.

[2] G.A. Gescheider, *Psychophysics: The Fundamentals*, Lawrence Erlbaum, 3 edition, May 1997.

[3] J. Zujovic, T.N. Pappas, D.L. Neuhoff, “Structural similarity metrics for texture analysis and retrieval,” *Proc. ICIP*, Nov. 2009, pp. 2225–2228.

[4] J. Zujovic, T.N. Pappas, D.L. Neuhoff, “Perceptual similarity metrics for retrieval of natural textures,” *Proc. IEEE Workshop Multimedia Sig. Proc.*, Oct. 2009.

[5] C.T. Meadow, *et al., Text information retrieval systems*, Emerald Group Publishing, 2007.

[6] X. Zhao, M.G. Reyes, T.N. Pappas, D.L. Neuhoff, “Structural texture similarity metrics for retrieval applications,” *Proc. ICIP*, Oct. 2008, pp. 1196–1199.

[7] J. Portilla, E.P. Simoncelli, “A parametric texture model based on joint statictics of complex wavelet coefficients,” *Int. J. Comp. Vis.*, vol. 40, p. 49–71, Oct. 2000.

[8] G. Wyszecki, W.S. Stiles, *Color Science: concepts and methods, quantitative data and formulae*, Addison-Wesley Publishing Co., 1982.

[9] S. Deshpande, S. Daly, “Synchronization mismatch: vernier acuity and perception evaluation for large ultra high resolution tiled displays,” *Human Vision and Electronic Imaging XV*, Jan. 2010, vol. 7527 of *Proc. SPIE*, pp. 75270L–1–12.

[10] “Corbis stock photography,” http://www.fotosearch.com/corbis/.

[11] U. von Luxburg, “A tutorial on spectral clustering,” *Statistics and Computing*, vol. 17, pp. 395–416, 2007.