Delft University of Technology

Training and Testing Texture Similarity Metrics for Structurally Lossless Compression

Zhang, Kaixuan; Shi, Zhaochen; Zujovic, Jana; De Ridder, Huib; Van Egmond, Rene; Neuhoff, David L.; Pappas, Thrasyvoulos N.

DOI
10.1109/TIP.2024.3364507

Publication date
2024

Document Version
Final published version

Published in
IEEE Transactions on Image Processing

Citation (APA)
Zhang, K., Shi, Z., Zujovic, J., De Ridder, H., Van Egmond, R., Neuhoff, D. L., & Pappas, T. N. (2024). Training and Testing Texture Similarity Metrics for Structurally Lossless Compression. *IEEE Transactions on Image Processing*, *33*, 1614-1626. https://doi.org/10.1109/TIP.2024.3364507

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Training and Testing Texture Similarity Metrics for Structurally Lossless Compression

Kaixuan Zhang, Zhaochen Shi, Jana Zujovic, *Member, IEEE*, Huib de Ridder, René van Egmond, David L. Neuhoff, *Life Fellow, IEEE*, and Thrasyvoulos N. Pappas, *Life Fellow, IEEE*

***Abstract*— We present a systematic approach for training and testing structural texture similarity metrics (STSIMs) so that they can be used to exploit texture redundancy for structurally lossless image compression. The training and testing is based on a set of image distortions that reflect the characteristics of the perturbations present in natural texture images. We conduct empirical studies to determine the perceived similarity scale across all pairs of original and distorted textures. We then introduce a data-driven approach for training the Mahalanobis formulation of STSIM based on the resulting annotated texture pairs. Experimental results demonstrate that training results in significant improvements in metric performance. We also show that the performance of the trained STSIM metrics is competitive with state of the art metrics based on convolutional neural networks, at substantially lower computational cost.**

***Index Terms*— Perceptual image quality, visual texture analysis, structural texture similarity metrics (STSIMs), matched-texture coding (MTC).**

I. Introduction

THE field of image compression has made significant advances over the past decades, relying on signal transformations, perceptual models, and lossless compression, to eliminate mathematical and perceptual redundancy. One of the keys to further advances is exploiting texture similarity. Current techniques have not been able to do this because they rely on quality metrics that are sensitive to point-by-point distortions that fail to account for the stochastic nature of texture and how it is perceived by humans [1], [2], [3], [4], [5]. To address such limitations, Zujovic et al. [4] introduced a new class of *structural texture similarity metrics (STSIMs)* that rely on region statistics, computed within each image, to evaluate texture similarity. Zujovic et al. [6] have identified three operating domains for evaluating the performance of texture similarity metrics: the ability to retrieve "identical" textures; the ability to distinguish between perceptually similar and dissimilar textures; and the ability to quantify distortion at the top of the similarity scale, that is, small deviations from identical textures. The performance of STSIMs in the first two domains, with applications to content-based retrieval, has been studied in [4] for identical and [6] for similar textures. The goal of this paper is to develop methods for training and testing STSIMs for applications that fall in the third domain, and in particular, for *structurally lossless* compression [5], whereby the original and compressed images may have visible differences in a side-by-side comparison, but have similar quality and one cannot tell which is the original. Structurally lossless compression is essential for exploiting texture self-similarity for image compression.

Traditional perceptual quality metrics rely on low-level properties of the human visual system [7], [8] to measure distortions near the threshold of perception. The goal is to achieve *perceptually lossless* compression, that is, images cannot be distinguished from the original in a side-by-side comparison at a given display resolution and viewing distance [8]. However, like peak-signal-to-noise ratio (PSNR), perceptual metrics still measure point-by-point distortions, albeit in the subband or DCT domain. This has prevented image (and video) compression algorithms from using spatial (and temporal) prediction for encoding textured regions, because large prediction errors can occur for textures that to the human eye appear to be equivalent. However, extended regions of texture (grass, clouds, foliage, concrete walls, fabric) could be simply replaced with previously encoded patches with similar statistics without any significant effect on perceived texture quality. The development of STSIMs has been the key for exploiting such texture redundancy, which naturally leads to the notion of structurally lossless image compression. STSIMs rely on region statistics, and can thus account for the typically stochastic nature of textures and the ability of the human visual system (HVS) to perceive textures with visible point-by-point differences as similar.

One approach for achieving structurally lossless compression is *matched-texture coding (MTC)* [9], [10], [11], [12]. It aims to exploit the self-similarity of an image, especially in

Manuscript received 25 July 2022; revised 17 April 2023 and 10 October 2023; accepted 31 December 2023. Date of publication 15 February 2024; date of current version 23 February 2024. An earlier version of this paper was presented at the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) [DOI: 10.1109/ICASSP.2012.6288145]. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Marc Antonini. *(Corresponding author: Thrasyvoulos N. Pappas.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Northwestern University Institutional Review Board, under Project No. 1071-003.

Kaixuan Zhang and Thrasyvoulos N. Pappas are with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208 USA (e-mail: t-pappas@northwestern.edu).

Zhaochen Shi was with the ECE Department, Northwestern University, Evanston, IL 60208 USA.

Jana Zujovic was with the ECE Department, Northwestern University, Evanston, IL 60208 USA. She is now with Google, Mountain View, CA 94043 USA.

Huib de Ridder and René van Egmond are with the Faculty of Industrial Design Engineering, Delft University of Technology, 2628 CE Delft, The Netherlands.

David L. Neuhoff is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA.

Data is available on-line at https://github.com/KaixuanZ/STSIM.

Digital Object Identifier 10.1109/TIP.2024.3364507

1941-0042 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: TU Delft Library. Downloaded on March 25,2024 at 12:02:35 UTC from IEEE Xplore. Restrictions apply.

textured regions. In MTC, selected image blocks are encoded by pointing to previously encoded blocks that are sufficiently similar; the remaining blocks are encoded by a baseline method, such as JPEG. Any image blocks can be encoded in this way, but the most significant gains are obtained when they contain textures, which require high bitrates when encoded by the baseline method. As a result, the success of MTC depends on a good texture similarity metric. To obtain such a metric, we propose a systematic approach for synthesizing texture variations that are encountered in image compression applications; we then design and conduct empirical studies for assigning perceived similarity scores to such variations; and use machine learning techniques for training STSIMs so that they can be used to exploit texture redundancy for image compression.

A key requirement for image quality metrics that are intended to be used both for quality assessment and as a tool within a compression algorithm is that they provide similarity measurements that are consistent with human perception. Agreement with subjective quality judgements can be evaluated by a number of statistical criteria, such as Pearson's linear correlation coefficient, Spearman's rank correlation coefficient ρ, and Kendall's rank correlation coefficient. The first measures the linear correlation between metric and subjective ratings, and the other two consider the relative rankings. However, as pointed out in [6], such quantitative assessment of image quality can only be achieved at the high end of the similarity scale; for severe distortions or dissimilar textures, the metric should simply give low similarity scores. For MTC coding in particular, it is important to find good matches for a (target) block of texture to be encoded and to order the candidates according to similarity to the target. To train a metric and to test its performance, we need a set of variations of a given texture and the associated perceived similarity scores. However, obtaining a set of images with fine similarity differences in the context of real compression applications (such as MTC) is difficult. Instead, we introduce a systematic approach for generating synthetic distortions that model variations that occur in natural textures, and conduct empirical studies for rating the similarity of the distorted and original textures. We then introduce a data-driven approach for training and testing the Mahalanobis formulation of STSIM based on the resulting annotated texture pairs. We also compare the performance of a number of existing variations of STSIMs that require no training (STSIM-1 and STSIM-2 [4]), or minimal training (STSIM-M [4] and STSIM-I [13]). Experimental results with an annotated dataset of 10 original grayscale texture images, three types of distortions (micro translations, micro rotations, and warping) and three degrees of each distortion demonstrate that the STSIMs provide excellent performance, but the data-driven approach can lead to considerable improvements. A set of 10 additional textures shows similar performance.

Gide et al. [14] conducted a similar empirical study using a comprehensive database they constructed as a benchmark for comparison of different texture quality metrics. Their database contains texture images with several types and degrees of distortions induced by traditional coding algorithms such as noise, blur, compression artifacts, shifts due to motion estimation, and synthesis based on a parametric texture model [15]. In contrast, our database construction and empirical studies are aimed at variations that occur naturally in textures, with the goal of exploiting texture self-similarity in compression applications.

Ding et al. [16] combined a VGG (Visual Geometry Group) convolutional neural network (CNN) [17] with SSIM [18] to develop DISTS (deep image structure and texture similarity), a data-driven image quality assessment technique that achieves excellent performance on texture similarity measurements and texture retrieval tasks. Experimental results with our annotated texture database demonstrate that the trained Mahalanobis formulation of STSIM is competitive with the performance of DISTS, trained on the same database, while offering a considerable advantage in computational efficiency over the VGG-based metric. In addition, we combine the two approaches by using a variant of the VGG CNN in place of the steerable filter bank in STSIM, and show that it matches the DIST performance.

The key ideas of this paper, including the construction of the database of geometric texture distortions and the basic experimental setup, were introduced in a conference paper [19]. For this paper, we modified the experimental setup to obtain subjective ratings that better reflect the perception of texture distortions. We also present extensive statistical analysis of the results of the two empirical studies, introduce a data-driven approach for metric training, and present extensive experimental results with an expanded set of textures. Overall, this paper provides a comprehensive presentation of the proposed approach with additional details, analysis, data, and experimental validation.

The remainder of this paper is organized as follows. Section II provides an overview of the basics of texture similarity metrics. In Section III, we describe the construction of the database of texture distortions. Two empirical studies to determine the similarity of the textures in the database are presented in Section IV and the data-driven approach for metric training is presented in Section V. The experimental results are discussed in Section VI, and the conclusions are summarized in Section VII.

II. Texture Similarity Metrics Basics

In this section we review grayscale texture similarity metrics. As we discussed, in order to take into account human perception and the (typically) stochastic nature of textures, STSIMs replace point-by-point comparisons with comparisons of region statistics. The idea of using region statistics was introduced by Wang et al. in the structural similarity (SSIM) metrics, which were implemented both in the space domain [18] and the complex wavelet domain [20]. However, SSIMs include a "structure" term that computes cross-image correlations, which are point-by-point comparisons, and as a result give low similarity values for perceptually similar textures. In contrast, STSIMs [4], [21], [22] only compare statistics computed within each image. The basic elements of STSIMs are:

Authorized licensed use limited to: TU Delft Library. Downloaded on March 25,2024 at 12:02:35 UTC from IEEE Xplore. Restrictions apply.

- A real or complex subband decomposition, such as the steerable filter decomposition.
- A set of statistics computed for each subband or pair of subbands of each image. For spatially varying images, the statistics are computed in a sliding window, and for homogeneous texture patches, as in this paper, they are computed over the entire subband. The statistics (mean, variance, horizontal and vertical autocorrelations, and crossband correlations) can be computed on the complex subband coefficients or their magnitudes.
- Formulas for computing similarity scores for each pair of corresponding statistics, one from each image. Different formulas may be used for different statistics, depending on the range of values that it takes, and may also include a normalization factor.
- A pooling strategy for combining the similarity scores, over statistics, subbands, and window positions, to produce an overall STSIM score.

Three variations, STSIM-1, STSIM-2, and STSIM-M, are presented in detail in [4]. All of these metrics use the complex steerable filters [23] to decompose the image into $n_s = 3$ scales and $n_o = 4$ orientations, plus a low frequency and a high frequency band. Since the low and high frequency bands are not split into orientations, the total number of subbands is $n_s \cdot n_o + 2 = 14$. All metrics compute the mean μ_x^m, variance σ_x^m, and horizontal and vertical autocorrelations $\rho_x^m(0,1)$ and $\rho_x^m(1,0)$ for each image or image patch x and each subband m, for a total of 56 statistics. STSIM-2 and STSIM-M also add crossband correlations $\rho_x^{m,n}(0,0)$, $n_s \cdot \binom{n_o}{2} = 18$ across all orientations for a given scale, and $n_o \cdot (n_s - 1) = 8$ across adjacent scales for a given orientation. Note that no crossband correlations are computed across the high frequency subband and the first scale and across the low frequency band and the last scale. Thus, STSIM-2 and STSIM-M use a total of $n = 82$ statistics. The statistics for STSIM-1 and STSIM-2 are computed on the complex subband coefficients, except for the crossband correlations that are computed on the magnitudes, while for STSIM-M all statistics are computed on the magnitudes [4].

STSIM-1 and STSIM-2 combine statistics in a fashion inspired by SSIM for an overall *similarity* score (details can be found in [4]). Like SSIM, these metrics do not satisfy the conditions of the mathematical definition of a metric [24], and thus, we refer to them as metrics in a loose sense. An alternative approach is to form an 82-dimensional vector of statistics $f(x)$ for each image or image patch, and to compute a *dissimilarity* score as the Mahalanobis distance between the feature vectors $f(x_1)$ and $f(x_2)$ for images x_1 and x_2, as follows [4]

$$D(x_1, x_2) = \sqrt{(f(x_1) - f(x_2))^H M (f(x_1) - f(x_2))} \quad (1)$$

where H denotes the Hermitian operator (complex conjugate transpose). It can be shown that the Mahalanobis distance satisfies the conditions of a proper metric, albeit in the 82-dimensional STSIM feature space rather than in the image space. The reason is that, if the features of two texture images are the same, the images are visually indistinguishable, but there is no guarantee that they are equal pixel-by-pixel. It can easily be shown that (1) satisfies the other two conditions of a metric, symmetry and triangular inequality.

Zujovic [4] used a diagonal Mahalanobis matrix M with the inverse of the variance of each feature, computed across all data, on the diagonal. Maggioni et al. [13] argued that, for image classification, the diagonal Mahalanobis matrix should use the intra-class variances, which reflect inherent texture variations in each class, so that inter-class variations can contribute to better discrimination among classes. We will refer to the resulting metric as *STSIM-I*.

Selecting a diagonal Mahalanobis matrix M assumes that the texture features $f(x)$ are independent, which may be a reasonable approximation, but is not necessarily true. In the following sections, we will discuss techniques for metric training, that is, designing the Mahalanobis matrix M. In particular, in Section V, we will present a data-driven approach for selecting the entries of the Mahalanobis matrix M.

In the past decade, CNN features have become popular for multiple tasks, such as style transfer [25], superresolution [26], and texture synthesis [27]. These techniques rely on CNN feature distance as a fidelity criterion. This has inspired the use of CNN features for image quality assessment (IQA) [28], [29], [30]. However, these methods were designed for images of faces, objects, and scenes, and do not reflect perceptual quality for texture images. As we discussed in the introduction, DISTS [16] is much more appropriate for texture quality. DISTS is based on a variation of the VGG CNN [17] and obtains the overall similarity of the images as a weighted sum of the luminance and structure terms of SSIM [18] computed for each feature. The task of machine learning is to determine these weights. Since DISTS performs well on IQA benchmarks and is also robust to texture distortions, the proposed proposed data-driven approach will be compared with DISTS in Section VI.

In addition to training the various STSIM metrics and DISTS, we will also combine the two approaches by using a variant of the VGG CNN in place of the steerable filter bank in STSIM. However, since the number of feature maps of the different layers is very large, we will only use means, variances, and horizontal and vertical first order correlations as features. We will not use any crossband correlations, because there are too many possibilities and no obvious way to select a smaller subset. We will refer to this metric as *STSIM-VGG*.

Finally, we should point out that all the metrics in this paper are not scale or rotation invariant. This is because in image compression and quality applications rotation and scaling are considered image distortions.

III. Database Construction

The performance of a texture similarity metric depends on how well it can quantify the perceived similarity of a pair of texture images. In this and following two sections, we propose a systematic approach for training and testing STSIMs to achieve this goal for image compression applications. In this section, we describe the construction of a database of texture distortions that model variations that occur in natural textures. In Section IV, we present empirical studies to determine

Authorized licensed use limited to: TU Delft Library. Downloaded on March 25,2024 at 12:02:35 UTC from IEEE Xplore. Restrictions apply.

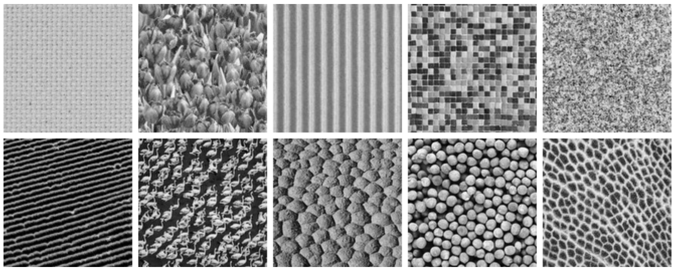


Fig. 1. Original texture images (1)-10, raster scan).

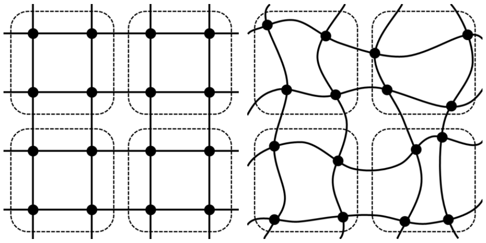
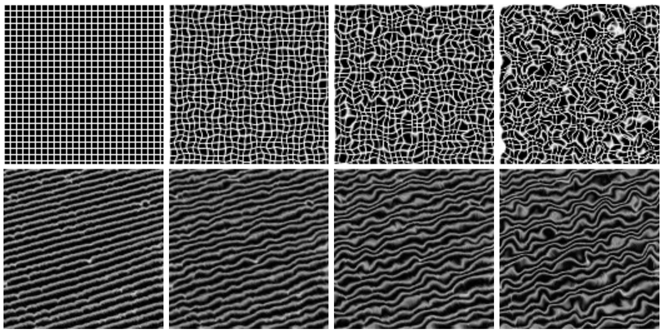


Fig. 2. Grid points in the original and distorted meshes.

the perceived similarity of the textures in the database, and in Section V, we present a data-driven approach for metric training, that is, designing the Mahalanobis matrix M, based on the annotated database.

To construct the database, we started by selecting a set of ten grayscale texture images, shown in Fig. 1, that range from random noise to highly structured textures. The resolution is 128×128 pixels. For each of these ten textures we generated three types of distortions, with three degrees of severity for each distortion. The first distortion consists of *random rotations* of small local patches. A $w \times w$ patch of the original texture is replaced by a patch of the original image rotated by an angle θ around the center of the patch. The values of θ are selected from a uniform distribution $U(-\theta_o, \theta_o)$. The second distortion consists of *random shifts* of small local patches. A $w \times w$ patch is replaced by a patch of the original image displaced by (dx, dy) from the patch location. The values of dx and dy are selected from uniform distributions $U(-D_x, D_x)$ and $U(-D_y, D_y)$. In both cases, the rotated or shifted patches are larger than the patch they replace so that there are no gaps in the image. The third distortion is *image warping*. The texture is warped according to random deviations of the control points of an underlying mesh, as illustrated in Figure 2. First, B-splines are fitted to the control points, and then bilinear interpolation is used to determine the pixel values. The software is available in MATLAB.[1] The control points are located on a $v \times v$ grid, and their deviations $(\delta x, \delta y)$ are selected from uniform distributions $U(-\Delta_x, \Delta_x)$ and $U(-\Delta_y, \Delta_y)$. Examples of the underlying meshes and the resulting distortions are shown in Figure 3.



(a) original (b) low dist. (c) medium dist. (d) high dist.

Fig. 3. Underlying meshes (top) and resulting warping distortions (bottom).

The severity of each type of distortion can be controlled by varying the parameters of the distributions of the rotation angles and the patch and control point displacements. As we discussed, these synthetic distortions are intended to model variations that occur in natural textures. In our experiments, we selected $w = 11$ for the patch rotations and translations and $v = 5$ for the mesh control points. The smaller meshes result in similar scale artifacts as those of the patches. This is because, as illustrated in Figure 2, four points on the grid control a rectangle shown in dashed lines, thus effectively producing deformations on rectangles of size equal to two times the grid spacing, with boundary smoothness and continuity conditions. Examples of the distorted images, corresponding to three different originals, are given in Figure 4. From left to right, we have three rotation-distorted images, three shift-distorted images, and three warped images, for a total of nine distortions for each texture. The severity of the distortions is increasing from left to right. For three levels of severity, we selected $\theta_0 \in \{0.3, 0.5, 0.7\}$ degrees for the random rotations, $D_x = D_y \in \{3, 5, 8\}$ pixels for the random shifts, and $\Delta_x = \Delta_y \in \{0.3, 0.5, 0.8\}$ pixels for the warping.

IV. Empirical Studies

In this section, we present two empirical studies to determine the perceived similarity of the textures in the database, first, to determine the similarity of the distortions of each texture *class* (original texture and associate distortions), and then to compare distortions across classes.

A. Study I: Within Texture Class

In the initial study [19], for each of the ten original textures, the participants were asked to rank the nine distorted textures on the basis of similarity with the original texture; tied ranks were not allowed. However, this procedure provides no indication of how close any of the distortions is to the original image. To better anchor the resulting perceptual scale with respect to the original texture, in the current study, we redid the experiments including the original in the ranking pool. A snapshot of the test is shown in Figure 5. The users were asked to drag and drop the ten images from the pink box into the gray boxes, according to increasing perceived distortion. The sequence of texture presentation was chosen randomly for each user, and the initial ordering of the ten images in the pink box was also random. There was a total of nine participants in this study; they were graduate students in the EECS department at Northwestern University and they all reported normal or corrected-to-normal vision. Each participant ranked ten groups of textures, one for each

[1] https://www.mathworks.com/matlabcentral/fileexchange/20057-b-spline-grid-image-and-point-based-registration

Authorized licensed use limited to: TU Delft Library. Downloaded on March 25,2024 at 12:02:35 UTC from IEEE Xplore. Restrictions apply.

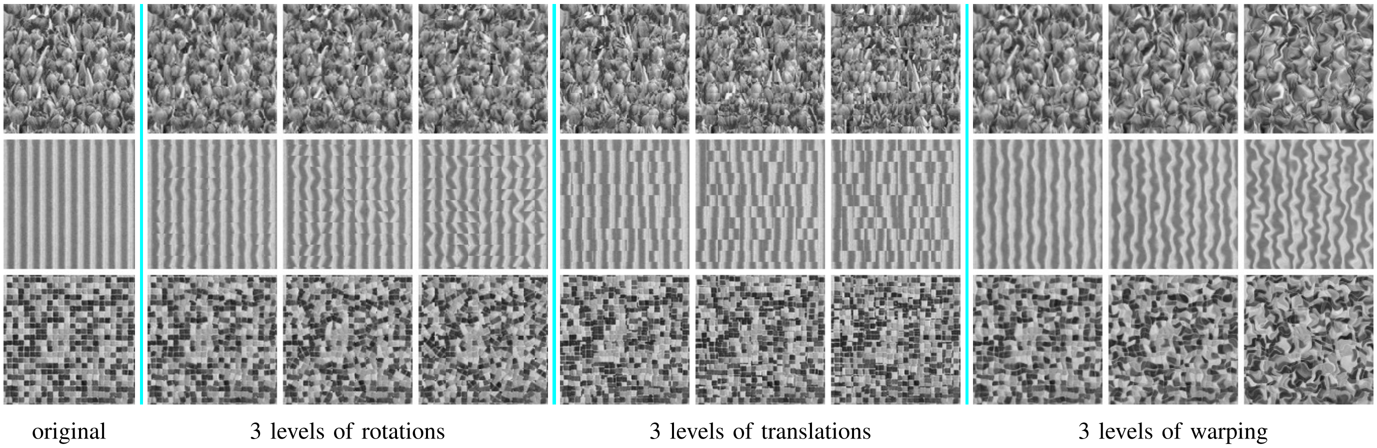
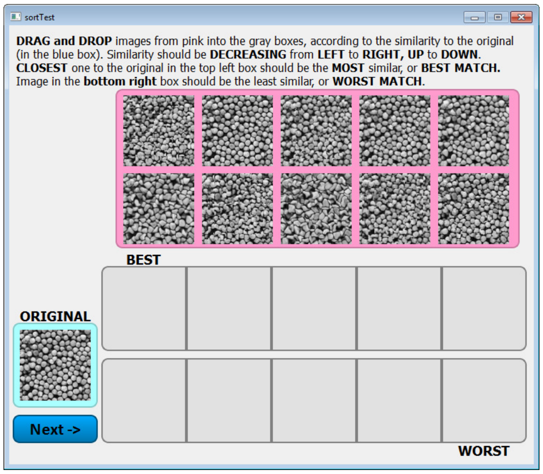


Fig. 4. Examples of distorted texture images.



DRAG and DROP images from pink into the gray boxes, according to the similarity to the original (in the blue box). Similarity should be **DECREASING** from **LEFT** to **RIGHT, UP** to **DOWN**. **CLOSEST** one to the original in the top left box should be the **MOST** similar, or **BEST MATCH.** Image in the **bottom right** box should be the least similar, or **WORST MATCH**.

Fig. 5. Snapshot of Test I: Sorting within texture class.

TABLE I
PARTICIPANTS THAT SELECTED ORIGINAL AS LEAST DISTORTED

texture	1	2	3	4	5	6	7	8	9	10
fraction	7/9	2/9	9/9	8/9	2/9	9/9	4/9	5/9	6/9	7/9

class. They conducted the test on their own machines, desktops or laptops with different resolutions, and were allowed to pick a comfortable viewing distance. Given the small size of the texture patches, in all cases, the resolution was adequate. The effective image resolution varied approximately from 20 to 30 cycles per degree. The participants were not told that one of the ten images they were asked to rank is the original image, and as Table I shows, they did not always select the original as the least distorted. The length of the test was about 15 minutes.

Once we have the ranking results, we need to convert them to ratings that reflect the perceived similarity between the original texture and the ten textures (the nine distortions and the original texture) it was compared with. This can be done in a number of different ways. In all cases, for each original image, we extract a ten-dimensional vector of ratings. This vector represents the ground truth with which the values of a similarity metric should be compared.

The simplest approach is to find the mean ranking across participants for each distorted image and to use that as its "subjective" position with respect to the original. This is perhaps one of the oldest techniques, proposed in 1770 by Jean-Charles de Borda, and today is known as Borda's rule. He called this method "election by order of merit," i.e., the cumulative preference given to a candidate is its final score.

In [19], Zujovic et al. tried two other popular approaches for analyzing this type of data, Thurstonian scaling [31], and multidimensional scaling (MDS) [32], [33], and found that the three approaches for analyzing the ranking data to obtain similarity ratings yield approximately the same results, so we used the simplest, Borda's rule. Moreover, Boschman [34], [35] shows that for categorical data the mean ranking results are comparable to the results obtained by Thurstonian scaling. The key question is whether the rankings can be considered as an interval scale instead of an ordinal one, and Boschman suggests that that is a fair assumption. In fact, we analyzed the ranking data using both the mean ranking and Thurstonian scaling and verified Boschman's conclusions.

Figure 6 shows the resulting ratings for each texture. The original texture is indicated as 0 and the three severity levels for each type distortion are indicated as 1, 2, and 3. Note that, with the exception of textures 2 and 8, the level 3 warping is perceived as the most distorted. However, the results very significantly from texture to texture for the other levels and distortion types. Again, note that the original is not always perceived as the least distorted (textures 2 and 5). Figure 7 shows the resulting order for each texture class. Note that the differences are most distinct for texture classes 3 and 6.

To ensure that we have meaningful results for metric training, we need to determine whether the differences between the distorted textures and the original are above the perceptual threshold. For this we used the *R-index* [36], [37]. According to Brown [36], the *R*-index is free of assumptions, is easy to calculate, and does not need a lot of replications. It is especially efficient for multiple comparisons where there is a reference image and several other images to be compared to the reference, and can be applied to ranking data where the reference is one of the test images, as is the case in our study. The *R*-index ranges from 50% (chance level) to 100%. Table II lists the *R*-index for each of the distortions of the

Authorized licensed use limited to: TU Delft Library. Downloaded on March 25,2024 at 12:02:35 UTC from IEEE Xplore. Restrictions apply.

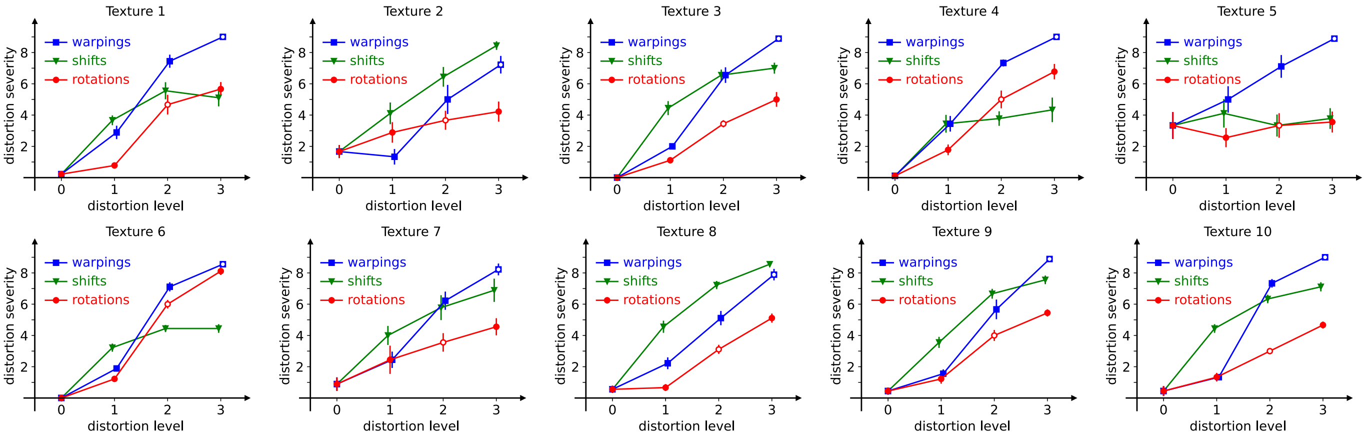


Fig. 6. Empirical Study I: Severity of distortions for each texture. Empty markers indicate distortions used in Study II. The error bars stand for standard errors, which are standard deviation divided by the square root of the number of participants.

TABLE II

R-INDEX FOR EACH DISTORTION OF EACH TEXTURE. RED NUMBERSINDICATE IMAGES BELOW THE DETECTION THRESHOLD

texture	1	2	3	4	5	6	7	8	9	10
original	50	50	50	50	50	50	50	50	50	50
rotation 1	77.8	67.3	100	96.9	42.0	100	68.5	54.9	74.1	76.5
rotation 2	100	82.1	100	100	50	100	90.1	98.7	99.4	94.4
rotation 3	100	85.8	100	100	51.9	100	94.4	100	100	100
shift 1	100	82.7	100	93.8	54.3	100	88.3	100	99.4	99.3
shift 2	100	95.7	100	99.4	51.2	100	95.0	100	100	100
shift 3	100	100	100	99.4	56.8	100	97.5	100	100	100
warping 1	100	39.5	100	99.4	69.1	100	79.0	90.7	82.7	85.2
warping 2	100	80.9	100	100	91.4	100	98.1	100	100	100
warping 3	100	99.4	100	100	100	100	100	100	100	100

10 texture classes. Given the number of distortions (9) and a preferred α-value of 0.05, which becomes 0.005 after Bonferroni correction, the threshold that the R-index is significantly different from 50% is $R = 78.36$. The table shows that most of the ratings are above threshold. However, the subthreshold data should also be taken into account in metric training, as the metric must yield low values for distortions that are below threshold.

B. Empirical Study II: Across Texture Classes

In order to obtain the subjective ranking of distortions *across* texture classes, in the final stage of the initial study [19], the participants were asked to rank the ten textures they selected as the most distorted in each class on the basis of similarity with the corresponding original texture. However, the data collected in this manner is very sparse, because each participant was asked to label only one subset of the distorted images out of 9^{10} possible subsets. Instead, in the current study, we asked the participants to rank *two* sets of distortions across the ten textures: the medium-level rotation distortions and the high-level warping distortions. This allows us to create a combined ordering of the subjective scores for all pairs of original and distorted textures, which can be converted to similarity ratings that can be used as ground truth for metric training and testing. A snapshot of this final stage of the test is given in Figure 8. This study was conducted by the same participants as the first study, on the same machines and viewing conditions. Each participant ranked two sets of distortions across the ten textures, one for the medium-level rotation distortions and one for the high-level warping distortions. The length of the test was about 5 minutes.

We used Borda's rule to convert the rankings of the textures for the two distortions to similarity ratings. The results are shown in Figure 9. A 2-way ANOVA showed that there was no significant interaction between texture and distortion type: F(9,160) = 1.90, p = 0.06, while there was a significant effect of texture: F(9,160) = 17.46, p < 0.001. We then combined the ratings of the two experiments. This was done in two stages. First, we scaled the within texture class distortions for each class from 0 for the original to 1 for distortion 33 (warping level 3). Let y^i be the scaled vector of distortions for each texture class i. This resulted in two negative values, which we later set to 0. We then solved a least squares problem to find the scaling value s_i for each texture class i that minimizes

$$\left(s_i\ y^i_{12} - z^i_{12}\right)^2 + \left(s_i\ y^i_{33} - z^i_{33}\right)^2 \qquad (2)$$

where z^i_{12} and z^i_{33} are the ratings for distortions 12 (rotation level 2) and 33 for the texture classes in the second experiment. The results are shown in Figure 10.

V. DATA-DRIVEN APPROACH FOR METRIC TRAINING

Among the different variations of STSIMs we reviewed in the previous section, we selected the Mahalanobis distance because it provides an intuitive combination of the different features (texture statistics) that can take into account their interdependencies and relative importance. STSIM-1 and STSIM-2 are not easily amenable to optimization. Moreover, they were developed with retrieval applications in mind, for example, in the selection of feature similarity formulas and the multiplicative combination of the similarity scores. Yet, as we will see, their performance is remarkably good without any training.

We take a data-driven approach for selecting the entries of the Mahalanobis matrix, in contrast to the diagonal matrix in [4] and [13], which assumes that the features are independent.

Authorized licensed use limited to: TU Delft Library. Downloaded on March 25,2024 at 12:02:35 UTC from IEEE Xplore. Restrictions apply.

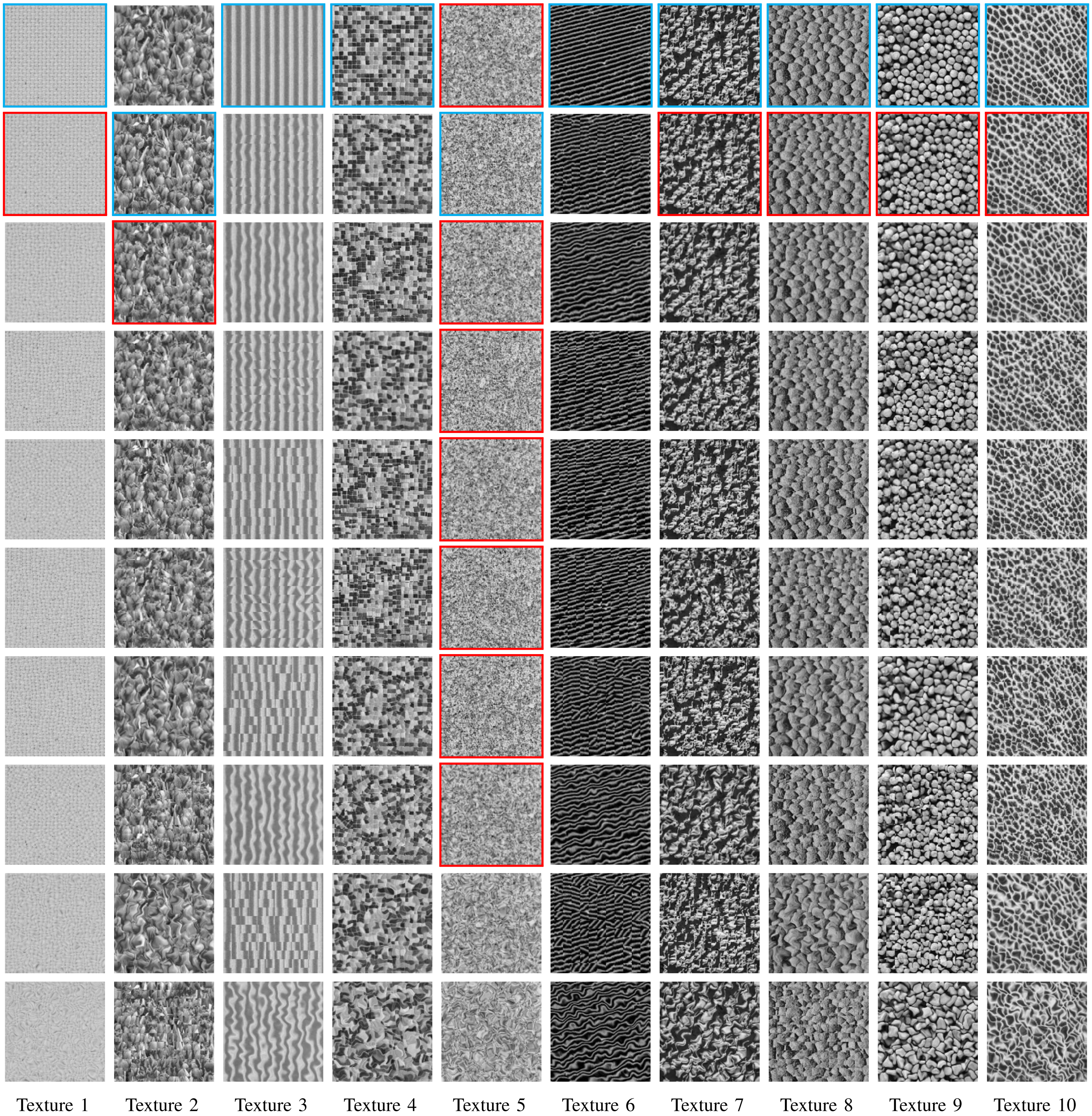


Fig. 7. Empirical Study I: Distortions ordered according to severity. Cyan box indicates original texture, red below threshold.

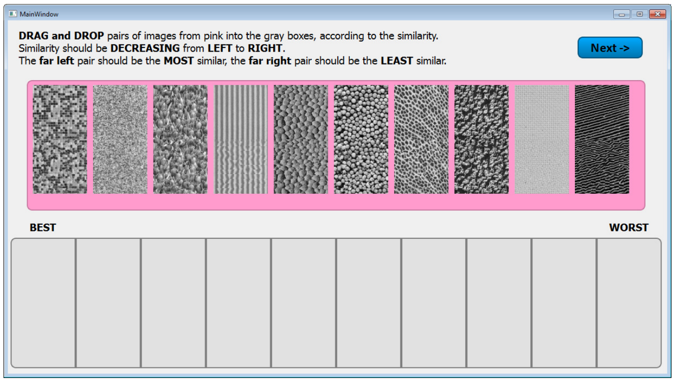
Since the $n \times n$ Mahalanobis matrix is positive semi-definite, it can be factored into a product $M = L^H L$, where L is an $r \times n$ matrix. Then we can rewrite (1) as follows

$$D_f(x_1, x_2) = ||L \cdot (f(x_1) - f(x_2))|| \tag{3}$$

where $||g|| = \sqrt{g^H g}$. We use the Adam optimizer to find the matrix L that maximizes the Pearson's (linear) correlation coefficient r (PLCC) between the ratings from the empirical test and the metric prediction. Details of the metric training will be introduced in V-A. The task is to estimate both the dimensionality of L and the values of its entries. We will refer to this formulation of the metric as *STSIM-Mf,* where f stands for factored. We will also use machine learning techniques to estimate the entries of a diagonal M, and will refer to the resulting metric as *STSIM-Md.*

An important implementation question that we will address in Section VI is the selection of the analysis filter bank. Zujovic et al. [4] used a complex steerable filter decomposition; for STSIM-1 and STSIM-2, the means, variances, and autocorrelations were computed with the complex subband coefficients, and the crossband correlations (STSIM-2) were computed with the coefficient magnitudes; for STSIM-M,

Authorized licensed use limited to: TU Delft Library. Downloaded on March 25,2024 at 12:02:35 UTC from IEEE Xplore. Restrictions apply.



DRAG and DROP pairs of images from pink into the gray boxes, according to the similarity.
Similarity should be **DECREASING** from **LEFT** to **RIGHT**.
The **far left** pair should be the **MOST** similar, the **far right** pair should be the **LEAST** similar.

Fig. 8. Snapshot of Test II: Sorting across texture classes (original texture at the top, medium-level rotation distortion at the bottom).

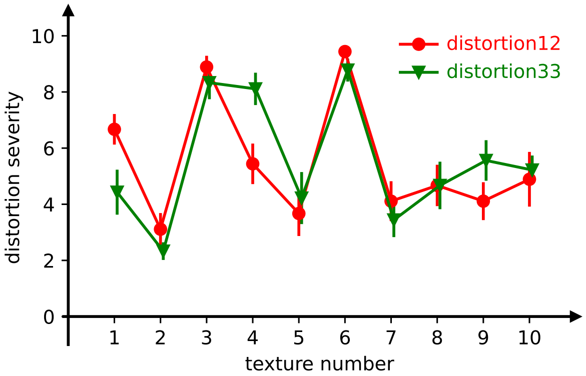


Fig. 9. Empirical Study II: Distortion severity across textures. The error bars stand for standard errors.

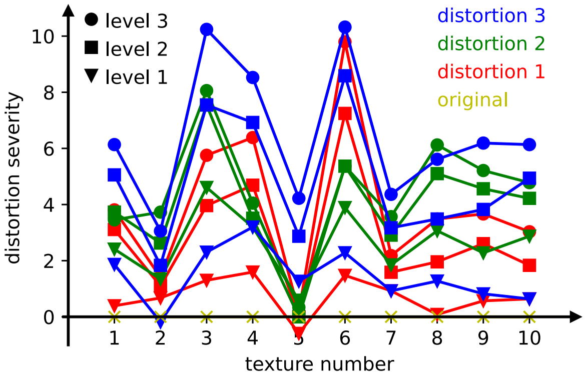


Fig. 10. Empirical Study II: Normalized distortion severity.

they found that magnitudes were most effective. However, the application domain of this paper is compression instead of retrieval, and their conclusions do not necessarily hold. We will consider both complex and real steerable filter decompositions. In addition, as we discussed in Section II, we will train STSIM-VGG, a variant of the VGG CNN in place of the steerable filter bank in STSIM, and appropriately selected features. Finally, we will compare the performance of the trained STSIM metrics with DISTS trained on the same data.

A. Metric Training

To compare the metric predictions with the empirical ratings, one can use PLCC, Spearman's rank correlation coefficient ρ (SRCC), or Kendall's rank correlation coefficient τ (KRCC). We based the metric training on PLCC maximization.

Let $\hat{y}_d^k$ be the metric prediction of the difference between original texture x_o^k and distorted texture x_d^k for class k given by (1):

$$\hat{y}_d^k = D(x_o^k, x_d^k) \tag{4}$$

We form a vector $\hat{\mathbf{y}}^k$ of the metric predictions for all distortions (including different seeds) of all types for class k, and a vector $\mathbf{y}^k$ of the corresponding empirical ratings. Then, the PLCC maximization is defined as follows:

$$\arg\max_{L} \ \ \text{PLCC}(\hat{\mathbf{y}}, \mathbf{y}) \tag{5}$$

with

$$\text{PLCC}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{K}\sum_{k=1}^{K} r(\hat{\mathbf{y}}^k, \mathbf{y}^k) = \frac{1}{K}\sum_{k=1}^{K} \frac{cov(\hat{\mathbf{y}}^k, \mathbf{y}^k)}{\sigma(\hat{\mathbf{y}}^k)\sigma(\mathbf{y}^k)} \tag{6}$$

where K is the total number of texture classes ($K = 10$), $\hat{\mathbf{y}}$ and $\mathbf{y}$ are vectors formed by concatenation of the vectors $\hat{\mathbf{y}}^k$ and $\mathbf{y}^k$ for all classes k, r is the PLCC between vectors $\hat{\mathbf{y}}^k$ and $\mathbf{y}^k$, and L is the Mahalanobis factor that appears in (3). If there are no perceptual ratings across classes, the PLCC is computed *locally*, within each class and then averaged across all classes, as in (6). If global perceptual ratings are available, then the PLCC is computed *globally*, across all available data, which are combined in two vectors $\hat{\mathbf{y}}^g$ and $\mathbf{y}^g$. Thus,

$$\text{PLCC}(\hat{\mathbf{y}}, \mathbf{y}) = r(\hat{\mathbf{y}}^g, \mathbf{y}^g) = \frac{cov(\hat{\mathbf{y}}^g, \mathbf{y}^g)}{\sigma(\hat{\mathbf{y}}^g)\sigma(\mathbf{y}^g)} \tag{7}$$

In the next section, we will show that global training leads to more robust metric performance.

We chose to optimize the metric by maximizing PLCC rather than SRCC or KRCC, because PLCC optimization is straightforward using a gradient descent approach, while the gradient calculation is not straightforward for the SRCC and KRCC approaches. Note that the typical optimization criterion used in machine learning is mean squared error (MSE)

$$\text{MSE}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{\dim(\hat{\mathbf{y}})} ||\hat{\mathbf{y}} - \mathbf{y}||^2 \tag{8}$$

which as we will show in Section VI, leads to suboptimal results. To optimize PLCC we minimize the negative correlation. The optimization is carried out with an Adam optimizer [38].

VI. Experimental Results

In this section, we present the results of metric training and compare the performance of the resulting metric with existing approaches. The performance of the texture similarity metrics will be based on comparisons with the values of perceived similarity that we obtained from the empirical studies we discussed in Section IV.

Authorized licensed use limited to: TU Delft Library. Downloaded on March 25,2024 at 12:02:35 UTC from IEEE Xplore. Restrictions apply.

TABLE III

EXISTING METRICS WITH STEERABLE FILTERS[a]

Criterion	PLCC		SRCC		KRCC	
Filters	real	complex	real	complex	real	complex
PSNR	0.349		0.399		0.284	
SSIM	0.426		0.465		0.336	
CW-SSIM	–	0.558	–	0.578	–	0.414
STSIM-1	0.640	**0.820**	0.703	**0.782**	0.520	**0.609**
STSIM-2	0.619	0.619	0.689	0.614	0.519	0.444
STSIM-M	0.608	0.629	0.556	0.732	0.418	0.556
STSIM-I	0.707	0.730	0.732	0.762	0.554	0.573

[a] There are no filters in PSNR and SSIM.

A. Existing Metrics

Zujovic et al. [19] relied on the results of the initial empirical study to compare the performance of PSNR, SSIM (space domain) [18], CWSSIM (complex wavelet domain) [20], and STSIM-1 and STSIM-2 [4]. The comparisons were based on PLCC and SRCC, and showed that the STSIMs outperformed the other metrics. They reported values for PLCC and SRCC that were computed within each class and averaged over all the texture classes.

In our experimental results we used the updated empirical studies, whereby the original texture image was included in the ranking pool in the within class study, and the medium-level rotation distortions and high-level warping distortions were used in the across class study. The combination of the two studies makes it possible to obtain a global similarity scale, and thus the performance criteria (PLCC, SRCC, and KRCC) can be computed globally. Table III shows the performance of existing data-independent metrics, PSNR, STSIM-1 and STSIM-2, and the data-dependent STSIM-M and STSIM-I. STSIM-1 and STSIM-2 were implemented as in [4], with the real/complex filter coefficients used for the correlation coefficients, and their magnitudes used for the crossband correlations. The data-dependence of STSIM-I and STSIM-M consists of estimating the variances on the diagonal, which was based on the same data that was used for the data-driven approaches. STSIM-M and STSIM-I were implemented as in [4], with the magnitude of the real/complex filter coefficients for all correlations. It is somewhat surprising that STSIM-1 has the best overall performance. As expected, STSIM-I performs better than STSIM-M.

B. Data-Driven Metrics

The total number of images that were used in our empirical studies is 100, 10 original images and nine distortions for each image. While the empirical study data is adequate for metric evaluations, a lot more data is required for metric training. To increase the amount of data without conducting cumbersome additional empirical studies, we generated 12 additional images for each distortion using the same parameters but different random seeds. We used the 1080 additional images for metric training and validation, and the initial 100 images for testing. We assumed that a different seed does not result in significant changes in texture appearance, so we used the same similarity ordering/ratings that we obtained in the empirical studies.

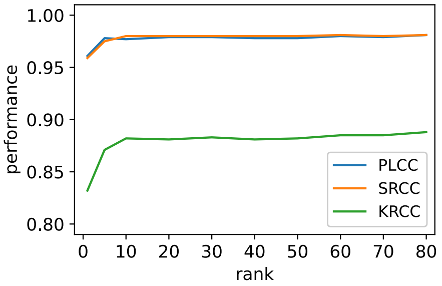


Fig. 11. STSIM-Mf performance for different ranks of M.

We trained the metrics using two different objective functions, MSE loss (8) and PLCC maximization (5). We selected PLCC maximization for the experiments reported in this section because it is the most straightforward to use. Since the values of the empirical ratings are normalized to the [0, 1] interval, while the Mahalanobis-based STSIM metrics take values in the $[0, \infty)$ range, the MSE loss requires the selection of a mapping of the metric values (or conversely the empirical ratings), which will affect the results.

We used the Adam optimizer [38] and 0.01 as the initial learning rate. Each metric was trained with 500 epochs. For STSIM-Mf based on complex steerable filters, we tried different values of l, the rank of the matrix L, covering the entire range from 1 to 82. To measure the goodness of fit, we used Pearson's r (PLCC), Spearman's ρ (SRCC), and Kendall's τ (KRCC), computed globally, as we discussed above. The results are shown in Figure 11. Observe that the performance increases sharply up to $l = 5$, and remains high after that, with the variations depending on the performance criterion. We selected $l = 10$ as a reasonable balance between performance and computational efficiency, as well as lower risk of over-fitting.

We first establish the importance of metric training using the global similarity scale. Figure 12 compares the performance of the data-dependent but minimally trained STSIM-M with the local and global optimization of STSIM-Mf. For the local optimization, we maximize the average of the PLCC computed within each class. In all cases, the metric values are plotted against the global perceptual ratings, and the performance criteria (PLCC, SRCC, and KRCC) can be computed globally. Note that local optimization aligns the data for each texture but fails to align the data across textures, as global optimization does, for a huge gain in performance. This is especially impressive in view of the fact that the behavior of two textures (2 and 5) deviates from the other ones. Figure 13 presents an alternative view of the resulting fit by comparing the rescaled original experimental data from Figure 10 with the global optimization. The ten panels, one per texture class, underscore how successful the training was. For example, the ratings for distortion type 1 (shifts) are not influenced by the level of distortion for textures 1, 4 and 6. This is correctly indicated by the model predictions.

We now compare the performance of the different data-driven metrics. Table IV shows the results of 5-fold cross-validation for the metrics trained with PLCC maximization. For the cross-validation we used the following grouping of the

Authorized licensed use limited to: TU Delft Library. Downloaded on March 25,2024 at 12:02:35 UTC from IEEE Xplore. Restrictions apply.

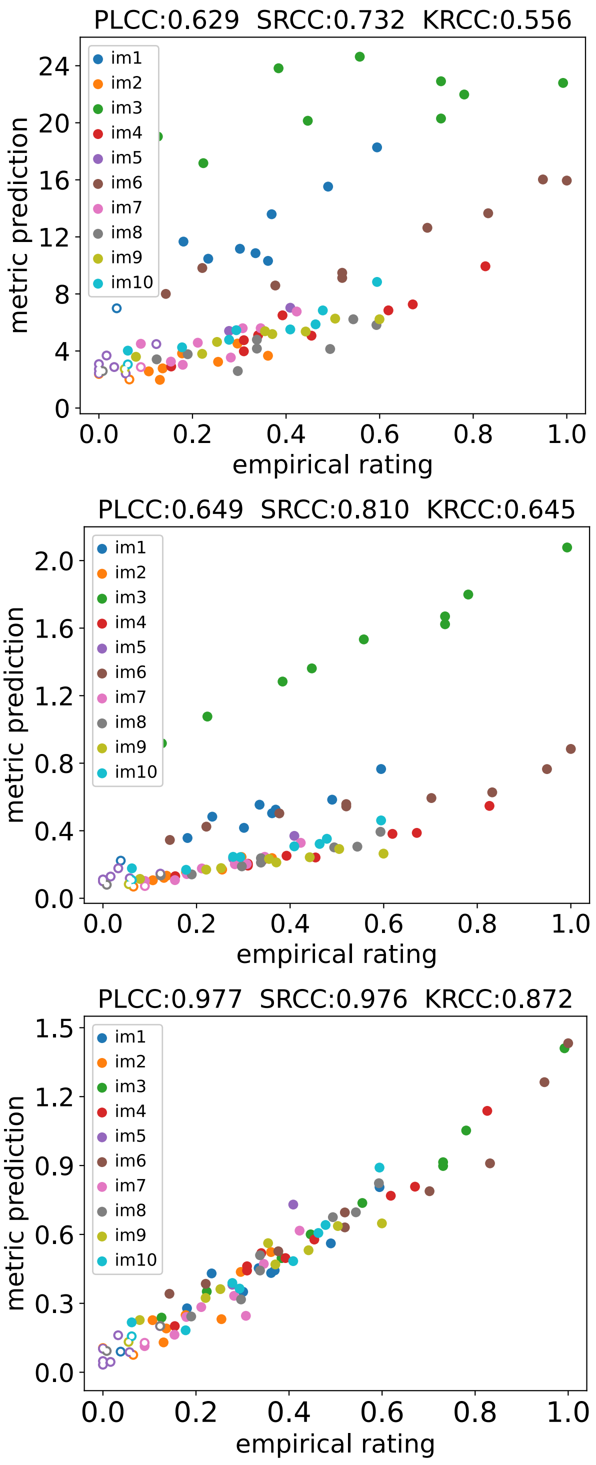


Fig. 12. Metric predictions for STSIM-M (top), local STSIM-Mf (middle), and global STSIM-Mf (bottom). Empty circles below threshold.

TABLE IV

5-Fold Cross Validation on the 10 Textures^c

Criterion	PLCC		SRCC		KRCC	
Filters	real	complex	real	complex	real	complex
STSIM-Md	0.726	**0.855**	0.710	0.859	0.573	0.721
STSIM-Mf	0.851	0.837	0.853	**0.874**	**0.731**	0.723
DISTS [16]	0.837		0.849		0.708	
STSIM-VGG	0.823		0.796		0.655	

^{c} DISTS and STSIM-VGG are based on the VGG CNN.

textures: (1,3), (2,4), (5,9), (6,10), and (7,8). The performance criteria for both training and testing were computed globally. STSIM-Md refers to the metric with diagonal Mahalanobis matrix trained with the data-driven approach. As we discussed, STSIM-Mf refers to the factored Mahalanobis matrix with $r = 10$. STSIM-VGG refers to STSIM combined with the VGG CNN in place of the steerable filters. Comparing with the results of existing metrics in Table III, we conclude that training improves performance, but it is not clear that adding terms off the diagonal helps. There is no clear conclusion on the relative performance using real and complex steerable filters. The performance of the trained STSIM metrics is slightly better than DISTS [16] and STSIM-VGG based on PLCC. The results vary when we consider SRCC and KRCC but the advantage of training is clear.

One important consideration is the batch size for metric training. STSIM-Mf has a small number of parameters, and thus, can be trained with very large batch size (all 900 textures and more). DISTS on the other hand is based on a neural net (VGG), which has a lot of parameters, and can only be trained with batch size of 60 on the latest GTX 3080. However, with global training and random shifting of the data, the batch size has no significant effect on the results.

C. Results With Additional Textures

To further test the robustness of the proposed techniques, we collected a set of 10 additional textures, shown in Figure 14. We will refer to the new textures as 11-20 in raster scan order. We conducted two empirical studies with a new group of 10 participants. In the first study, we asked the participants to rank the distortions of the new set of textures on the basis of similarity with the corresponding original texture. In the second study, we asked the participants to rank the medium-level rotation distortions and the high-level warping distortions across all 20 textures. The studies were conducted in the same manner as the studies in Section IV.

First, we applied the metrics that were trained on the 10 textures of Figure 1 to the new set of textures. The results are shown in Table V. The performance of the trained STSIM metrics has dropped compared to the 5-fold cross validation of Table IV, indicating that they are not as robust as DISTS, which now has the best overall performance. However, the results clearly demonstrate the advantage of metric training. The drop in performance can be attributed to the fact that the textures in the second set are different. A more accurate evaluation of performance can be obtained by cross-validations, which we consider next.

We then conducted a 5-fold cross-validation test using all 20 textures. We used the following random grouping of the textures for the test: (2,6,8,18), (10,14,16,19), (3,12,13,17), (4,5,7,15), (1,9,11,20). The results, shown in Table VI are comparable to the original cross-validation results of Table IV. Note the impressive performance of STSIM-VGG, which is comparable to DISTS.

Authorized licensed use limited to: TU Delft Library. Downloaded on March 25,2024 at 12:02:35 UTC from IEEE Xplore. Restrictions apply.

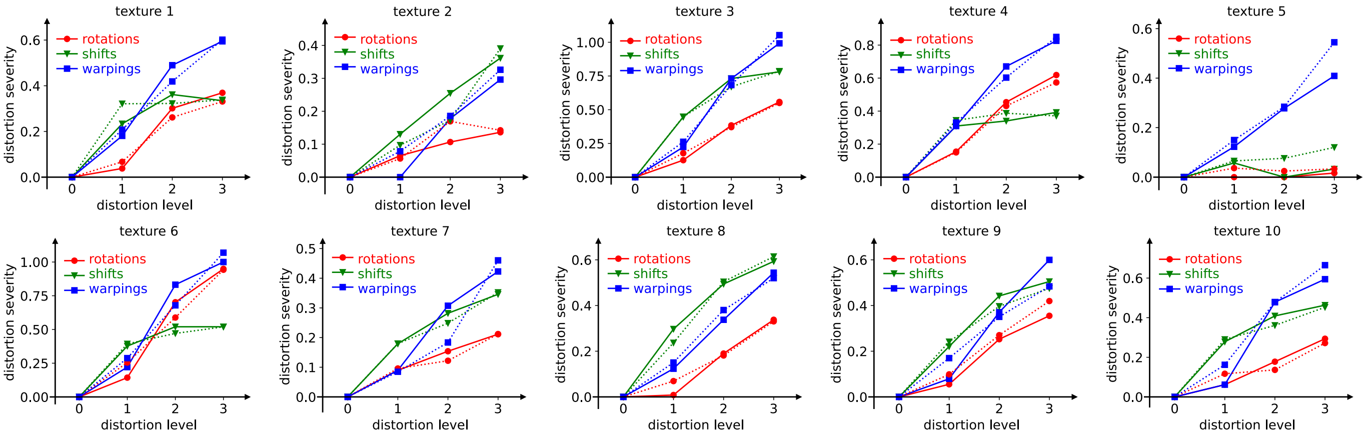


Fig. 13. Final results: Normalized distortion severity (solid lines) and metric predictions (dotted lines).

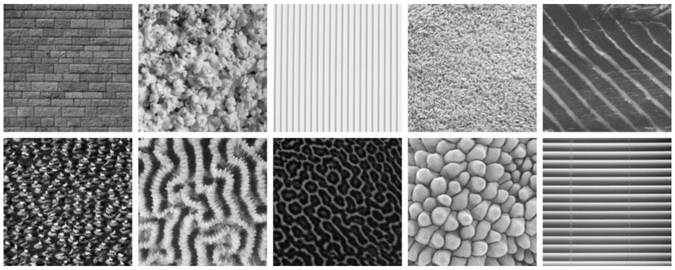


Fig. 14. Additional texture images (11-20, raster scan).

TABLE V
TRAIN ON FIRST SET OF 10 TEXTURES, TEST ON SECOND SET

Criterion	PLCC		SRCC		KRCC	
Filters	real	complex	real	complex	real	complex
PSNR	0.182		0.151		0.100	
SSIM	0.354		0.383		0.265	
CW-SSIM	–	0.566	–	0.541	–	0.386
STSIM-1	0.631	0.719	0.679	0.669	0.494	0.483
STSIM-2	0.702	0.656	0.771	0.525	0.577	0.391
STSIM-M	0.514	0.741	0.362	0.814	0.259	0.620
STSIM-I	0.740	0.745	0.687	0.800	0.509	0.609
STSIM-Md	0.801	0.791	0.784	0.829	0.597	0.638
STSIM-Mf	0.764	0.779	0.741	0.773	0.557	0.579
DISTS [16]	**0.834**		0.841		**0.652**	
STSIM-VGG	0.787		**0.842**		0.642	

TABLE VI
5-FOLD CROSS-VALIDATION ON ALL 20 TEXTURES

Criterion	PLCC		SRCC		KRCC	
Filters	real	complex	real	complex	real	complex
STSIM-Md	0.726	0.825	0.749	0.842	0.573	**0.679**
STSIM-Mf	0.755	0.823	0.764	0.821	0.591	0.653
DISTS [16]	0.844		**0.849**		0.677	
STSIM-VGG	**0.846**		0.845		0.667	

D. Image Quality

As we discussed in the Introduction, the goal of this paper is to develop metrics for structurally lossless image compression [5]. While developing a method that incorporates the proposed metrics is beyond the scope of this paper, we used the metric to evaluate the quality of the results presented in [9] and [10]. Figure 15 shows an original image compressed with JPEG and MTC [9]. The similarity based on the proposed metric is shown in Table VII for different sliding window sizes with 50% overlap. The metric values reflect the fact that the JPEG image has visible distortions in the neck and hands of the woman, while the MTC image has no obvious artifacts, even though the sweater texture is a bit different than the original.

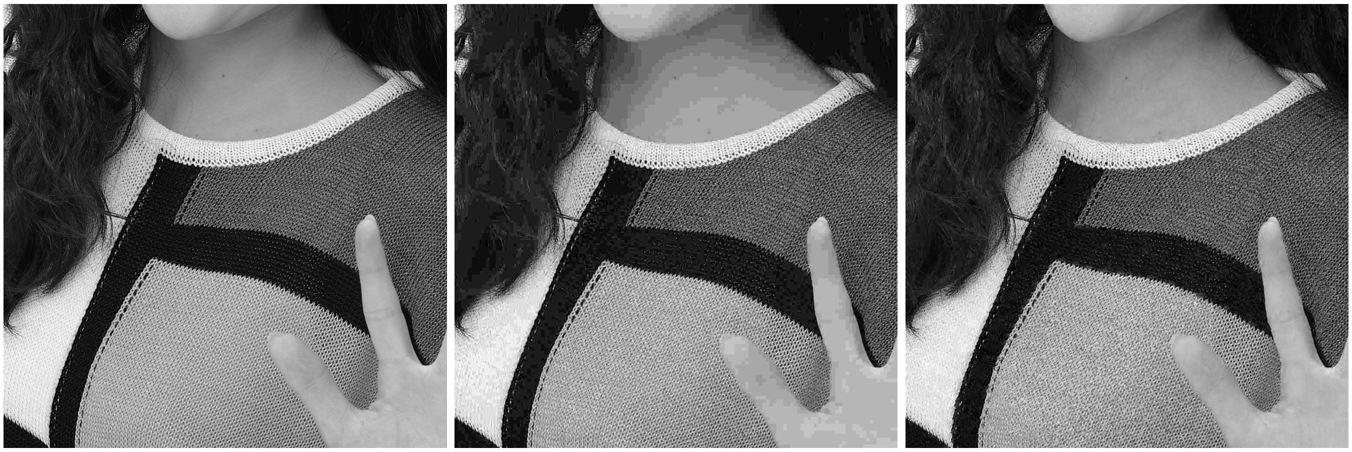
TABLE VII
STSIM-MF TESTED ON THE IMAGES IN FIGURE 15

Block Size	128x128		256x256		512x512	
Overlap	0%	50%	0%	50%	0%	50%
JPEG	0.529	0.499	0.398	0.412	0.353	0.351
MTC	0.453	0.443	0.352	0.349	0.295	0.310

E. Computational Cost

The experiment results above show that the trained STSIM is competitive with DISTS in terms of performance. However, STSIM metrics have significant computational advantages. The first advantage is the use of a (relatively compact) filter bank. VGG has thousands of feature maps (convolution kernels), while the steerable filter decomposition has only 14 subbands. Since convolution is the most computationally intensive operation in the metric, STSIM-Mf is dramatically faster for both training and inference. Another advantage is the reuse of feature vectors. For the Mahalanobis-based metrics (STSIM-Mf and STSIM-VGG) the statistics (features) are computed independently for each texture image, while the structure term in DISTS requires a pair of images as input. Thus, for STSIM-Mf and STSIM-VGG, we only need to compute the feature vectors once, and save them for future training and inference, while for DISTS the features must be recomputed for each new pair of textures. For instance, in a texture retrieval task with N query images, and M reference images, STSIM-VGG requires $N+M$ forward propagations of VGG to generate and store the feature vectors for all images, whereas DISTS requires $2NM$ forward propagations of VGG to compute the feature vectors for all pairs of query and reference images.

In terms of actual running time, the implementation of the complex steerable filters is actually slower because VGG is a widely used network that has been optimized very well



Original

JPEG

Matched-Texture Coding (MTC)

Fig. 15. Image compression at 0.34 bits/pixel [9].

in pytorch. Thus, DISTS and STSIM-VGG have comparable speeds of 4.6 ms/frame, compared with 28 ms/frame STSIM-Mf. On the other hand, the steerable filters use considerably fewer GPU resources. For training 45 images in a batch, VGG requires 8 GB of GPU memory, while the steerable filters require 2 GB of GPU memory.

VII. Conclusion

We presented a systematic approach for training and testing structural texture similarity metrics (STSIMs) so that they can be used to exploit texture redundancy for structurally lossless compression. Accordingly, the training and testing is based on a set of image distortions that reflect the characteristics of the perturbations present in natural texture images. We conducted empirical studies to determine human perception of the severity of these distortions and used them to train STSIMs. Our experimental results show that training significantly improves metric performance. We have also demonstrated that the performance of the trained STSIM metrics is competitive with state of the art metrics based on convolutional neural networks trained on the same data, at substantially lower computational cost. Future research will include a broader range of image deformations, and incorporation of the trained metrics in matched texture coding [9], [10] and other structurally lossless compression algorithms.

References

[1] P. Ndjiki-Nya, T. Hinz, and T. Wiegand, "Generic and robust video coding with texture analysis and synthesis," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2007, pp. 1447–1450.

[2] A. C. Brooks, X. Zhao, and T. N. Pappas, "Structural similarity quality metrics in a coding context: Exploring the space of realistic distortions," *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1261–1273, Aug. 2008.

[3] T. N. Pappas, J. Zujovic, and D. L. Neuhoff, "Image analysis and compression: Renewed focus on texture," *Proc. SPIE*, vol. 7543, pp. 178–189, Jan. 2010.

[4] J. Zujovic, T. N. Pappas, and D. L. Neuhoff, "Structural texture similarity metrics for image analysis and retrieval," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2545–2558, Jul. 2013.

[5] T. N. Pappas, D. L. Neuhoff, H. de Ridder, and J. Zujovic, "Image analysis: Focus on texture similarity," *Proc. IEEE*, vol. 101, no. 9, pp. 2044–2057, Sep. 2013.

[6] J. Zujovic, T. N. Pappas, D. L. Neuhoff, R. van Egmond, and H. de Ridder, "Effective and efficient subjective testing of texture similarity metrics," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 32, no. 2, p. 329, Feb. 2015.

[7] M. P. Eckert and A. P. Bradley, "Perceptual quality metrics applied to still image compression," *Signal Process.*, vol. 70, no. 3, pp. 177–200, Nov. 1998.

[8] T. N. Pappas, R. J. Safranek, and J. Chen, "Perceptual criteria for image quality evaluation," in *Handbook of Image and Video Processing*, 2nd ed., A. C. Bovik, Ed. Cambridge, MA, USA: Academic Press, 2005, pp. 939–959.

[9] G. Jin, Y. Zhai, T. N. Pappas, and D. L. Neuhoff, "Matched-texture coding for structurally lossless compression," in *Proc. 19th IEEE Int. Conf. Image Process.*, Orlando, FL, USA, Sep. 2012, pp. 1065–1068.

[10] G. Jin, "Matched-texture coding for structurally-lossless image compression," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Northwestern Univ., Evanston, IL, USA, Jun. 2016.

[11] G. Jin, T. N. Pappas, and D. L. Neuhoff, "An adaptive lighting correction method for matched-texture coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 2006–2010.

[12] G. Jin, T. N. Pappas, and D. L. Neuhoff, "Improved side matching for matched-texture coding," in *Proc. Eur. Wksp. Vis. Info. Proc. (EUVIP)*, Paris, France, Dec. 2014, pp. 1–5.

[13] M. Maggioni, G. Jin, A. Foi, and T. N. Pappas, "Structural texture similarity metric based on intra-class variances," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Paris, France, Oct. 2014, pp. 1992–1996.

[14] M. S. Gide and L. J. Karam, "On the assessment of the quality of textures in visual media," in *Proc. 44th Annu. Conf. Info. Sci. Syst. (CISS)*, 2010, pp. 1–5.

[15] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 49–71, Oct. 2000.

[16] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2567–2581, May 2022.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scele image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[19] J. Zujovic, T. N. Pappas, D. L. Neuhoff, R. van Egmond, and H. de Ridder, "Subjective and objective texture similarity for image compression," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 1369–1372.

[20] Z. Wang and E. P. Simoncelli, "Translation insensitive image similarity in complex wavelet domain," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Philadelphia, PA, USA, Feb. 2005, pp. 573–576.

Authorized licensed use limited to: TU Delft Library. Downloaded on March 25,2024 at 12:02:35 UTC from IEEE Xplore. Restrictions apply.

[21] X. Zhao, M. G. Reyes, T. N. Pappas, and D. L. Neuhoff, "Structural texture similarity metrics for retrieval applications," in *Proc. 15th IEEE Int. Conf. Image Process.*, San Diego, CA, USA, Oct. 2008, pp. 1196–1199.

[22] J. Zujovic, T. N. Pappas, and D. L. Neuhoff, "Structural similarity metrics for texture analysis and retrieval," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Cairo, Egypt, Nov. 2009, pp. 2225–2228.

[23] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multiscale transforms," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 587–607, Mar. 1992.

[24] D. Brunet, E. R. Vrscay, and Z. Wang, "On the mathematical properties of the structural similarity index," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1488–1499, Apr. 2012.

[25] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423.

[26] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision—ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 694–711.

[27] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in *Proc. Conf. Neural Inf. Process. Syst.*, vol. 28. Cambridge, MA, USA: MIT Press, 2015, pp. 262–270.

[28] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 586–595.

[29] E. Prashnani, H. Cai, Y. Mostofi, and P. Sen, "PieAPP: Perceptual image-error assessment through pairwise preference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 1808–1817.

[30] S. Bosse, D. Maniry, K. Müller, T. Wiegand, and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 206–219, Jan. 2018.

[31] L. L. Thurstone, "A law of comparative judgment," *Psychol. Rev.*, vol. 34, no. 4, p. 273, 1927.

[32] W. S. Torgerson, *Theory and Methods of Scaling*. New York, NY, USA: Wiley, 1958.

[33] J. B. Kruskal and M. Wish, *Multidimensional Scaling*. Beverly Hills, CA, USA: Sage, 1977.

[34] M. C. Boschman, "ThurCatD: A tool for analyzing ratings on an ordinal category scale," *Behav. Res. Methods, Instrum., Comput.*, vol. 32, no. 3, pp. 379–388, Sep. 2000.

[35] M. C. Boschman, "DifScal: A tool for analyzing difference ratings on an ordinal category scale," *Behav. Res. Methods, Instrum., Comput.*, vol. 33, no. 1, pp. 10–20, Feb. 2001.

[36] J. Brown, "Recognition assessed by rating and ranking," *Brit. J. Psychol.*, vol. 65, no. 1, pp. 13–22, Feb. 1974.

[37] H. Lee and D. Van Hout, "Quantification of sensory and food quality: The R-index analysis," *J. Food Sci.*, vol. 74, no. 6, pp. 57–64, Aug. 2009.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA, May 2015, pp. 1–15.

Kaixuan Zhang received the B.S. degree in electrical engineering from East China Normal University, Shanghai, China, in 2017, and the M.S. degree in electrical engineering from Northwestern University, Evanston, IL, USA, in 2019, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include image processing, computer vision, and machine learning.

Zhaochen Shi received the B.S. degree in telecommunications engineering with management from the Beijing University of Posts and Telecommunications, Beijing, China, in 2019, and the M.S. degree in electrical engineering from Northwestern University in 2021. His research interests include image processing, computer vision, and machine learning.

Jana Zujovic (Member, IEEE) received the Diploma degree in electrical engineering from the University of Belgrade, Belgrade, Serbia, in 2006, and the M.S. and Ph.D. degrees in electrical engineering and computer science from Northwestern University, Evanston, IL, USA, in 2008 and 2011, respectively. She was a Post-Doctoral Fellow with Northwestern University from 2011 to 2013. She is currently a Senior Staff Software Engineer with Google, Mountain View, CA, USA. Her current research interests include image and video analysis, image quality, and multiview 3D geometry.

Huib de Ridder received the M.Sc. degree in psychology from the University of Amsterdam, Amsterdam, The Netherlands, in 1980, and the Ph.D. degree in technical sciences from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 1987. He is currently an emeritus Professor in informational ergonomics with the Delft University of Technology, The Netherlands. From 1982 to 1998, he was affiliated with the Vision Group, Institute for Perception Research, Eindhoven, and from 1998 to 2021 he was with the Department of Industrial Design in Delft. His chair concerned human information processing, focusing on visual perception, and decision making. In 2008, he (co-)founded the Perceptual Intelligence Laboratory. His current research interests include the fields of art and perception and medical imaging.

René van Egmond received the Ph.D. degree from the Nijmegen Institute for Cognition and Information, Radboud University, in 1996. He was a Postdoctoral Research Fellow with The Ohio State University from 1996 to 1997 and a Postdoctoral Fellow of the Dutch Science Organization (NWO) from 1997 to 2000. In 2000, he joined the Department of Industrial Design Engineering, Delft University of Technology. His research interests include product sound design and perception and human information processing. He teaches classes on interactive audio design, embodied audio design, and cognitive ergonomics for complex systems. He has conducted research projects in cooperation with external partners, such as Philips, Toyota, European Space Agency, and in external funded projects by the European Union.

David L. Neuhoff (Life Fellow, IEEE) received the B.S.E. degree from Cornell University, Ithaca, NY, USA, in 1970, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 1972 and 1974, respectively. He served as a Faculty Member of the University of Michigan, Ann Arbor, MI, USA, from 1974 until his retirement in June 2019, as the Joseph E. and Anne P. Rowe Emeritus Professor in electrical. From 1984 to 1989, he was an Associate Chair of the EECS Department and again from 2008 to 2018. He spent two sabbaticals with Bell Laboratories, Murray Hill, NJ, USA, and one with Northwestern University, Evanston, IL, USA. His research and teaching interests include communications, information theory, and signal processing, especially data compression, quantization, image coding, image similarity metrics, source-channel coding, sensor networks, and Markov random fields. He co-chaired the 1986 IEEE International Symposium on Information Theory. He was the Technical Co-Chair of the 2012 IEEE Statistical Signal Processing Workshop and initiated the IEEE Information Theory Society effort to install statues of Claude Shannon around the USA. He has been an Associate Editor for IEEE TRANSACTIONS ON INFORMATION THEORY (twice). He has served on the Board of Governors of the IEEE Information Theory Society (twice) and he was its President in 2006.

Thrasyvoulos N. Pappas (Life Fellow, IEEE) received the S.B., S.M., and Ph.D. degrees in electrical engineering and computer science from MIT in 1979, 1982, and 1987, respectively. From 1987 to 1999, he was a member of the Technical Staff with Bell Laboratories, Murray Hill, NJ, USA. In 1999, he joined the ECE Department, Northwestern University. His research interests include human perception and electronic media, and in particular, image quality and compression, image analysis, content-based retrieval, model-based halftoning, and tactile and multimodal interfaces. He is a fellow of SPIE and IS&T. He has served as the Editor-in-Chief for IEEE TRANSACTIONS ON IMAGE PROCESSING, the Vice President-Publications for the Signal Processing Society (SPS) of IEEE, and the Co-Chair of the SPIE/IS&T Human Vision and Electronic Imaging Conference. He is currently the Editor-in-Chief of the *Journal of Perceptual Imaging* (IS&T).

Authorized licensed use limited to: TU Delft Library. Downloaded on March 25,2024 at 12:02:35 UTC from IEEE Xplore. Restrictions apply.