Volume xx (200y), Number z, pp. 1–11

Texture Browser: Feature-based Texture Exploration

Xuejiao Luo Leonardo Scandolo Elmar Eisemann

Delft University of Technology, The Netherlands.

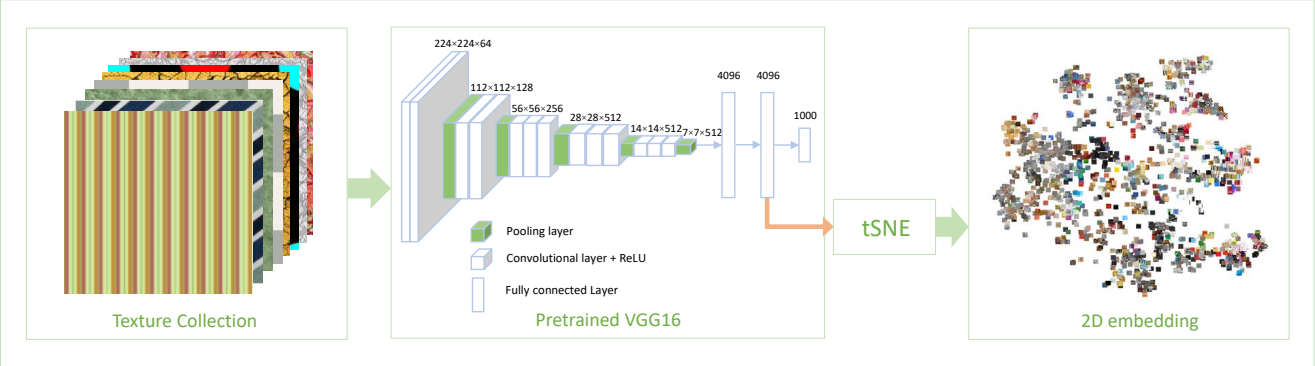

Figure 1: *Workflow of our manifold embedding: each image in a texture collection is run through the VGG16 network to obtain a feature vector extracted from the penultimate layer of the network. These features are then embedded in 2D space using the t-SNE algorithm.*

Abstract

Texture is a key characteristic in the definition of the physical appearance of an object and a crucial element in the creation process of 3D artists. However, retrieving a texture that matches an intended look from an image collection is difficult. Contrary to most photo collections, for which object recognition has proven quite useful, syntactic descriptions of texture characteristics is not straightforward, and even creating appropriate metadata is a very difficult task. In this paper, we propose a system to help explore large unlabeled collections of texture images. The key insight is that spatially grouping textures sharing similar features can simplify navigation. Our system uses a pre-trained convolutional neural network to extract high-level semantic image features, which are then mapped to a 2-dimensional location using an adaptation of t-SNE, a dimensionality-reduction technique. We describe an interface to visualize and explore the resulting distribution and provide a series of enhanced navigation tools, our prioritized t-SNE, scalable clustering, and multi-resolution embedding, to further facilitate exploration and retrieval tasks. Finally, we also present the results of a user evaluation that demonstrates the effectiveness of our solution.

CCS Concepts

• ***Human-centerd computing*** → *Visualization; Visualization systems and tools; Visualization toolkits;*

1. Introduction

Texture is a fundamental visual characteristic that reflects the surface detail of an object, usually stored in an image. It plays an important role in object recognition in computer vision [GRM*18], rendering in computer graphics [Hec86], and artistic or industrial design [LP15]. Most importantly, texture is recognized easily by an observer [Gib50]. Nowadays, huge texture image collections exist, both natural and computer-generated but retrieving an ideal texture can be very difficult.

Traditional texture browsing methods tend to utilize metadata, such as keywords, captions, or descriptions, to manually cluster samples into groups. In this context, a vocabulary of 47 texture terms was created to describe a large collection of natural textures [CMK*14, BRL97]. This vocabulary tries to relate commonly-used texture words in English to the visual properties of textures, describing a wide variety of texture patterns. Such metadata-driven retrieval systems have several drawbacks. First, it requires considerable effort and time to manually annotate samples to build a database. Second, it is extremely difficult for a layman to describe the content

of different types of textures in words. Finally, differences in language systems and bias on an individual's interpretation make it hard to reach a universal standard for properly and precisely describing a texture.

Alternatively, a texture can be procedurally generated, for example by modeling the process to form a texture from texel samples [WL00], or labels [DWL*19]. This process requires expertise and it is difficult to ensure a good exploration. Recently, a semi-procedural approach [GAD*20] has been presented, working with an exemplar and a label map to then create new variations. The chosen input is crucial to the process and a label map is not always available. Even in dedicated solutions for material generation, such as Substance [Sub], browsing for input textures remains a common task, which we address.

In this paper, we propose an efficient visual navigation tool that can organize an unlabeled texture collection in a semantically meaningful way. This enables users to efficiently navigate to a target texture. Potentially, this can also ease the task of labeling large collections of textures with semantic terms if wanted.

In our framework, we utilize the activations of the penultimate dense layer of a pre-trained image recognition network, VGG16, to represent the input images [SZ14]. This high-dimensional activation vector represents multi-scale, and thus semantically-meaningful, features of the input and has been shown to link to texture information [OMS17]. To visualize the distribution of textures based on these feature vectors, we use t-SNE to embed them into two dimensions. For easier navigation, we propose a modification of the original t-SNE algorithm, which we call prioritized t-SNE, which can modify the resulting 2D embedding to facilitate exploration of a chosen image neighborhood.

We also propose several navigation methods including flip zooming, scalable mean-shift clustering, user interactive selection, and an image-based retrieval algorithm to accelerate texture retrieval, as shown in a user evaluation. As texture structure is usually independent of its color palette, recoloring [CFL*15] is a very convenient way to achieve diverse and plausible alternatives of the selected target and is integrated as well for the completeness of our system.

Compared to traditional state-of-the-art image retrieval systems, our framework targets texture images and offers a simple, user-friendly, and efficient interface for retrieval, navigation, and recoloring. More importantly, it does not require any manual preprocessing or labeling. This is a key point, as labeling abstract textures is a hard problem, requiring the use of specific language that is not easily interpreted by end-users. Our system organizes and displays textures in a way that greatly expedites retrieval and exploration tasks.

The main contribution of our work is an intuitive retrieval and exploration system for large unlabeled collections of texture images that spatially groups textures sharing similar features. Our system integrates several targeted tools, and while it builds upon some known techniques (often to leverage familiarity, e.g., selection tools and 2D interaction), it also introduces novel solutions to control the embedding.

The rest of the article is organized as follows. We fist review the related work (Sec. 2), before elaborating on our approach (Sec. 3), including the embedding and navigation solutions, as well as the proposed interface (Sec. 4). We then evaluate our work and compare to existing methods (Sec. 5) before we conclude (Sec. 6).

2. Related Work

An abundance of work focuses on image retrieval but most is not geared specifically towards textures. Metadata-based image retrieval systems have been successfully applied in most web-based image search engines [Tan19]. Nevertheless, these methods can still produce a lot of unwanted search results.

Color-based retrieval methods compute color histograms to guide the search task [SS*16, CBGM97] but are not focused on the actual content of the texture. Nevertheless, in our context, the structure of the images plays a much larger role and color can even be adapted in a post-process. Content-based image retrieval (CBIR) [PG17] relies on combinations of colors, textures, local geometry, or any other information that represents and can be extracted from images. In this situation, it is common to rely on example images when searching, which are not always available and these systems do not lend themselves well to a fine-grained search. Different metrics can be used to compare color and texture features [YLLF11] and many measures have been proposed in the past [Jul62, Yel93, MM96, PS00, DC14, ZPN13, ZRPN08, WBSS04]. Kokare et al. [KCB03] explore the impact of different similarity measures in distance-based automated retrieval tasks .

Text-based queries are very common but not easily applicable to texture search. This also holds true when involving a hybrid image retrieval system that relies on keywords and images [DAK10] or an interactive browsing solution [Por06], due to the large amount of manual effort in creating the text or image descriptions, and also due to the difference in each individual's interpretation. We compare to such an approach in our user study [CMK*14]. Sketch-based image retrieval methods utilize sketching to query target images [LSS*17]. While very powerful when focusing on important lines, e.g., when navigating sketch collections, not all properties of textures can be well captured. It is nonetheless a useful element in case that pronounced features are available and is also employed in our system.

With the tremendous success of convolutional neural networks (CNNs) on the ImageNet data set, image classification has received great attention [KSH12], [SLJ*15], [SZ14]. These systems learn complex features that outperform hand-crafted ones. Hand-crafted features (e.g., SIFT [Low04] and Gaussian mixture models [PFJ03]) were not used since available pre-trained network features have proven more versatile in recent years [ZYT17, LCF*19]. Their use for the task of object search within an image collection has been explored with promising results [BSCL14, XHZT15, GV19], but not oriented towards texture retrieval. Danon et al. [DAEFCO19] proposed an unsupervised learning method towards a metric of similarity, leveraging the fact that the similarity of two patches can be learned from the prevalence of their spatial proximity in natural images, which does not hold for textures with spatially homogeneous structures.

Good examples in image retrieval with deep networks exist [MMB*17, GARL17, YHNYD15, NAS*17, ZYT17] but require specific inputs, i.e., query images or keywords, and they do not target

navigation/exploration. When a database of labeled images is available, networks can be trained for other applications as well, such as recognition [XZD18], segmentation [FACO17], or procedural texture generation [HDR19]. Solutions to better address perceptual texture properties exist [H*12, PHRC13] as well but these systems are based on similarity matrices generated by capturing subjective human judgments. This is time-consuming, costly, and would have to be applied for each newly inserted item. This limits their generalization to large collections or different databases. Our solution is inspired by these approaches, as we leverage the powerful semantic feature extraction capabilities of deep neural networks in combination with modern visual data analysis.

Visual exploration of embedded datasets using t-SNE is used extensively in the medical field, for example for mass cytometry data exploration [LvUH*18]. Moreover, exploration methods based on hierarchical stochastic neighbor embedding [PHL*16] have been proposed [HPvU*17, HVP*19]. Nevertheless, in these cases, the focus is the exploration of relations between data points, which does not translate well to texture database exploration and retrieval tasks. The levels of the HSNE hierarchy are a subset of the dataset, and thus using such representation would hide information from the user. Our prioritized t-SNE approach follows a similar goal, has a simpler definition that does not require any preprocessing, continuously adjusts the relevance of each embedded point, and does not remove any images from view.

In the context of interactive exploration and browsing of image collection, previous work [H*12, PHRC13, Por06] proposed several visualization concepts, such as cylinder displays. However, these approaches are mostly intended for gaining an overview. Coarse semantic information-based methods are proposed to enhance the visualization and exploration. Yang et al. [YFH*06] and Mizuno et al. [MWT14] propose to use multidimensional scaling (MDS) [CC08] when computing the similarity between images for visualization, but either keywords are involved in the annotation and search process [YFH*06] or the resulting embedding is less efficient than t-SNE [MWT14]. MDS was further applied with weights to emphasize items of interest [DWF*18], while in Projection Explorer for Images [POM07, ENP*09] (PEx-Image) it was used for image embedding. Worring et al. [WK15, WKZ16] use pivot tables to visualize and explore the multimedia images over user-supplied metadata. To utilize the higher-level semantic features of the image, Xie et al. [XCZ*18] propose to train an image captioning model based on existing semantic keywords. Previous work dealing with metadata agnostic CBIR systems includes the work of Rodden et al. [RBSW99], who propose a grid visualization of images based on an MDS arrangement with color histograms of image sections as features. Tian et al. [TT00] propose a series of tools to create a virtual reality system for CBIR, based on a weighted MDS distribution over manually crafted color and texture features, as well as available text metadata. Gomi et al. [GMIL08] introduce a hierarchical CBIR system, with a hierarchical arrangement of images based on aggregated color values of image regions, and also available keywords associated with each image. Finally, Schaefer et al. [Sch16] present an overview of different image collection browsing methods, mostly designed for the case of photography collections, highlighting their own sphere and honeycomb arrangement methods based on an MDS embedding over median color information. As absolute color is not a suitable feature for texture browsing, these methods are not well suited for our application context.

3. Our Approach

Our work aims at facilitating navigation to a desired texture within a large database without any metadata. To this end, we position the textures in a 2D layout, following an embedding of their high-dimensional feature vectors, and propose interaction mechanisms to explore the dataset. The workflow is illustrated in Fig. 1. To support navigation, we provide solutions to influence the embedding at multiple scales. Features can be emphasized by allowing them to span a larger portion of space or irrelevant regions can be shrunk in real time. Additionally, a user can make use of clustering and image-based search (using existing images or sketches) to narrow down the search for texture structures. Finally, the colors of the target can be adapted to meet the wishes of the user. In the following, we describe the details of our solution.

3.1. Embedding

As discussed in Sec. 2, CNNs have proven to be a very powerful tool for extracting features from images. Typically, the latent vector that results from the penultimate layer of a network encodes relevant features that are important for classification. While it might sound attractive to build a specialized CNN for an image collection, it is a very expensive and time-consuming task, and suitable labels for training are typically not available.

Motivated by previous work in the area [OMS17, SZ14, GRM*18], we use (pre-trained) VGG16, which is a deep convolutional image classification network that was trained with ImageNet [FF10], a large image database. Evidence has been presented that deep convolutional image-classification networks trained on ImageNet rely on identifying texture rather than shape [GRM*18] and are therefore well suited for our tasks. The specialization of its descriptors for texture recognition is also leveraged in style-transfer approaches [GEB16]. The first layers of such networks encode filters for simple patterns, and later layers recognize progressively more complex textures [OMS17]. Nevertheless, the last layers are strongly linked to the actual categories recognized by the network and are therefore less useful for our application. Therefore, we selected the descriptors of the penultimate fully connected layer given their generality and efficacy. As an alternative, we also tested the use of layers from AlexNet [KSH12] but found them to be less effective in differentiating texture. We believe this is because its shallower architecture and large receptive fields force early layers to specialize more strongly on the recognition categories. In contrast, VGG16 is deeper and has smaller receptive fields, which recognize small abstract patterns at early layers which combine in later layers to recognize more complex patterns [OMS17].

We scale all input images to a resolution of 256×256 to provide as an input to the network. The second fully connected layer, which is the optimal layer for retrieval [YYLZ18], outputs a latent vector of 4096 dimensions, which we associate with each image.

The high dimensionality of the feature vector makes it impossible to directly visualize an organization of the images in such space.

For visualization and interaction purposes on a 2D screen, a two-dimensional positioning is most suitable. We use t-SNE [MH08], a dimensionality-reduction method, to embed the latent vectors. Fig. 2, shows examples of the resulting embedding for the Describable Textures Dataset (DTD) [CMK*14] and the UIUC Texture Database [Kaz16]. t-SNE has the advantage of being non-linear and preserves neighborhood relations, which is particularly desirable for exploration purposes. Within a collection, some textures may manifest as transformations of others, such as rotation, scaling, or coloring. While it would be possible, we do not enforce transform invariance, and thus they are not necessarily mapped to similar locations by the t-SNE algorithm. This is by choice, since these aspects can play into a semantic meaning (e.g., tiger stripes are typically vertical).

Although non-convex and non-deterministic, the embedding of t-SNE is efficient, fairly stable, and has proven useful in many applications. The tunable perplexity parameter was set to 30 heuristically for all our examples, based on the recommended range (5-50). Moreover, the t-SNE algorithm is run for a maximum of 2000 iterations. Both the computational and the memory complexity of t-SNE are $O(n^2)$ but optimized implementations exist [CRHC18, PTM*20].

3.2. Prioritized t-SNE

The original t-SNE formulation attempts to create an embedding that maintains similar neighborhoods as in the high-dimensional space. It does so by minimizing the Kullback-Leibler divergence between the joint probabilities of the distances of the low-dimensional embedding and the high-dimensional vectors [MH08]. Instead, we would like to offer the possibility for a user to emphasize certain groups of textures (or features) that should then make use of more space in the embedding in a continuous manner to avoid generating artificial clusters. Similarly, textures that are of less relevance should use less space. To achieve this, we propose the prioritized t-SNE embedding. We assume that each of potentially multiple user-selected samples (textures) S has a weight $W \in [0,10]$. The embedding region around this sample will be enlarged if $W \geq 1$ and shrunk if $0 \leq W < 1$. For a single selected sample S, we define weights w_i for all other samples as:

$$w_i = 1 - (1-W)e^{-d_i * max\left(W, \frac{1}{W}\right)} \tag{1}$$

where d_i is the distance between sample i and the selected sample S. When several samples are selected, the weights produced for each of the samples (Eq.1) are multiplied. Once the weights are determined, we compute the joint probabilities q_{ij} of map point y_j and map point y_i in the low-dimensional space by

$$q_{ij} = \frac{(1 + w_i * w_j * ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||)^{-1}}. \tag{2}$$

The embedding then uses standard t-SNE with these modified joint probabilities. To accelerate convergence, we initialize q_{ij} with the originally computed t-SNE counterpart. Fig. 3 shows an example using two selected samples.

3.3. Multi-scale Replication

Some textures manifest different visual features at different scales, as shown in Fig. 5. The t-SNE algorithm clusters such images based on an overall dominating feature, while a user might have been interested in a feature at a different scale. For example, imagine a piece of cloth with a very fine structure that forms a larger-scale pattern. On the one hand, it could be grouped with cloth textures. On the other hand, its large-scale pattern might be more suitable to be represented by other textures. Especially large-scale (i.e., low-frequency) features can have an important impact, since a user will initially see an overview of the collection, where the images are small, and mostly large-scale features are visible. In this sense, we would like an image to be embedded adjacent to images sharing its small-scale visual features, but also adjacent to images sharing its large-scale features. To resolve this conflict, we create additional embedding positions for each image that correspond to image versions where small-scale features are removed, and we replicate the images at those positions when necessary. Specifically, for each image we generate two blurred versions via convolution with a Gaussian kernel, approximating human feature perception, with a standard deviation of 5 and 9 pixels ($\sim 2\%$ and $\sim 3.5\%$ of the input resolution). These kernel sizes were empirically chosen, since smaller kernels had little impact, and larger ones quickly resulted in converging feature vectors for all images. Using several scales enables a successive removal of small-scale features (see Fig. 5). After obtaining three sets of latent vectors (original texture collection and the two blurred versions), we embed these with t-SNE in parallel.

In this case, we initially get an embedding with three times the image samples of the original collection. Nevertheless, the 2D embedding of the different versions for most images ($\sim 95\%$) remains very similar. This means that the original latent vector for these images is dominated by the large-scale image features. To avoid unnecessary clutter, we only replicate images when their distance in the 2D embeddings exceeds a given threshold distance (25% of the embedding diagonal), as the texture then clearly exhibits distinct multi-scale features. If not, only the original texture is kept, as it already well reflects its large-scale features on its own.

3.4. Clustering

While the presented embedding can successfully group similar textures, it fails to give a good overview due to too much information being displayed at once, especially with a large database. In consequence, we propose a clustering mechanism to ease navigation and to allow the user to identify categories in the embedding to then focus on clusters reflecting the desired properties, e.g., stripe patterns. To produce clusters, we apply Mean-Shift clustering [FH75] on the embedded feature space. This clustering algorithm has the advantage that it is guided by a single parameter, the scale at which the elements are grouped. Further, as the mean-shift can be computed relatively quickly by discretizing the space [BEDT08, HPvU*16], the scale can be chosen interactively. Given clusters of similar textures, we can enhance them with a representative image (*landmark*); the image nearest to the centroid of each cluster. Fig. 4 shows three examples of clustering visualization using landmarks for 6, 10 and 20 clusters. The center landmark image is surrounded by eight more images, which are representatives for the farthest texture in the

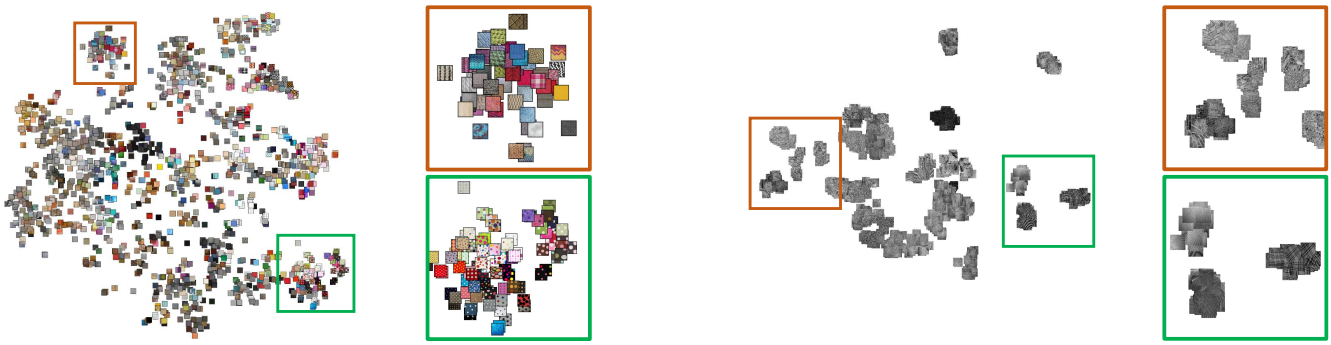

Figure 2: *The 2D embedding distribution of the DTD collection (left) and the UIUC collection (right). The colored sections zoom in some clusters in the distribution.*

Figure 3: *Original embedding (left) and modified embedding using prioritized t-SNE (right). The texture circled in red was given a higher priority with $W = 8$ in Eq.1, resulting in larger distances between images for all similar images, which are also enlarged to improve visibility. Conversely, the image circled in blue was given a low priority with $W = 0.2$ in Eq.1, contracting similar images together.*

corresponding direction (horizontal, vertical, or diagonal) of the embedding that still belongs to the cluster, as seen from the centroid. Hereby, the observer gets an idea of the texture appearance at the boundaries of the cluster.

4. Interface

The interface of our retrieval system is shown in Fig. 6. It consists of our two major components: the semantic embedding window and the priority texture selection tool. Further, we see an image-based search tool with an optional sketching board, and a recoloring tool. The latter is not a novel technical contribution, but is present for completeness. For our tests, we use the textures in the DTD collection. It is a database with 1518 images, grouped via 47 words (terms/categories) inspired from human perception.

The semantic embedding window (Figs. 6, 7) presents the collection of images to the user, positioned according to the computed embedding. This provides a holistic view of the image set and shows the global semantic transition among textures of different styles. The user can freely navigate through the overall display and change its zoom levels with the mouse. The texture that is pointed at with the mouse cursor is displayed in a large tooltip for a better detail visualization. In this window, the user can additionally enable the display of clusters to help obtain an overview of the features in different parts of the embedding (Fig. 7, top right). The user can select the number of clusters and their landmarks and representative images of the clusters are shown (Sec. 3.4). Furthermore, the user can select a cluster or a rectangular region to restrict the view to only the chosen subset of images. For this restricted view, the user can then opt for a flip-zooming visualization style, or a tiled visualization, to prevent image overlap; this is shown in the bottom row of Fig. 7. Lastly, as we support recoloring, images can be displayed in grayscale to avoid user bias based on color. Nevertheless, loss of contrast during this conversion can hide important texture details, which is why both options exist. Note that a detail-preserving decolorization scheme (e.g., [AAB11, AA16]) could be applied as future work.

The priority texture selection tool allows a user to select one or more images, and assign a priority to them, which is then used to update the embedding using our prioritized t-SNE (Sec. 3.2). This updated embedding will show images related to high-priority selections at a larger scale and images related to low-priority ones at a smaller scale, as shown in Fig. 3.

The image-based search tool is added for completeness of the system and allows a user to search based on a user-provided image, provided via a file or as a sketch. The simple sketch tool supports controllable pen color and width. We rely on VGG16 to extract the corresponding latent vector of the user image, and find the closest texture image in feature space. Fig. 8 shows an example of user sketches and the corresponding result in the DTD collection. Once a texture has been identified in this way, it is possible for the user to highlight it in the semantic embedding window as an additional landmark (Sec. 3.4) or to change its weight to influence the embedding.

The recoloring tool (Fig. 9) allows a user, as a final step, to change the color of a selected texture with the subsequent option to save it to a new file for use outside of our application. The recoloring algorithm is optimized based on [CFL*15] to simplify user interac-

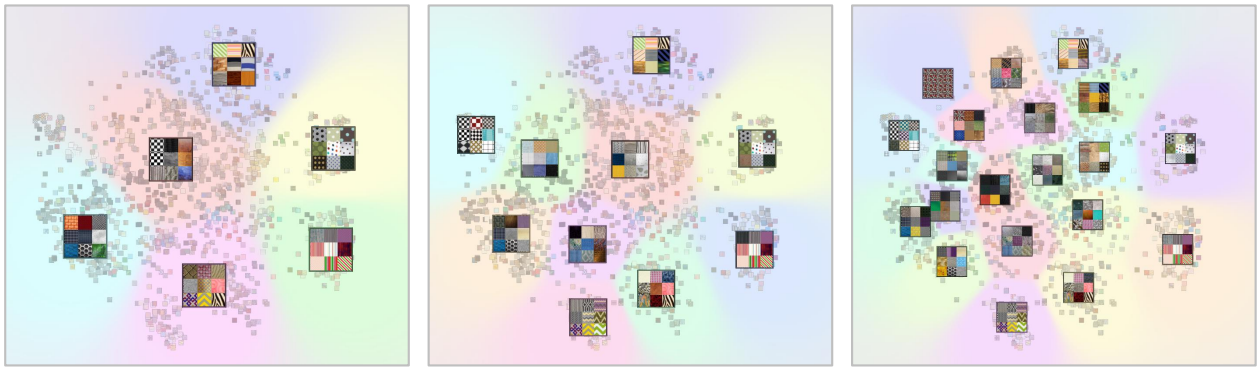

Figure 4: *Overview using mean shift on the 2D embedding distribution resulting in 6 (left), 10 (middle) and 20 (right) clusters. Landmarks summarize the cluster information by showing representative cluster images, and background color indicates cluster extent.*

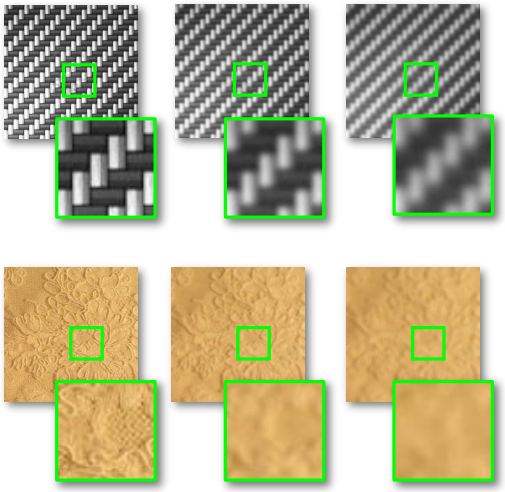

Figure 5: *Features at different scales (original and 5, 9 pixel blur). Low scale detail is successively removed.*

tion. We cluster the colors in the original image using a k-means algorithm, where we let the user select the number of clusters. The color of each cluster center is shown to the user, who can use a color picker to change them. Once a new cluster color is selected, we transfer the same color offset to all cluster pixels.

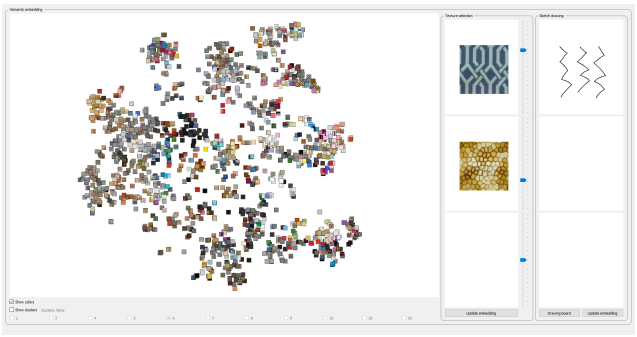

Figure 6: *Image-retrieval Interface. The images in their semantically embedded positions (left). Controls for weighing selected images for our prioritized t-SNE (middle) and a sketch interace (right).*

Figure 7: *A selection can be shown as interactive flip-zoom interface (left) or a tiled interface (right).*

5. Evaluation

We conducted a user evaluation to validate the effectiveness and advantages of our method. We wish to perform two evaluations: comparison with alternative methods, and features evaluation. In the first part of our study, we compared against displaying images on a grid (using thumbnails), which is the standard file-system solution. It is the visualization that users typically have the most experience in using. It also requires no metadata (which also holds as is the case for our system). Additionally, we compare to text-based solutions [CMK*14]. We also evaluate the four features compared to only using our basic *overview* mode, *multi-scale replication*, *clustering*, *prioritized-tSNE*, and *image-based search*. The first two features aim at general use cases and evaluation of them requires no special settings of the target. Therefore, they were integrated into the first part of the study, acting as additional comparisons. For the latter two features, they aim to improve the retrieval in cases

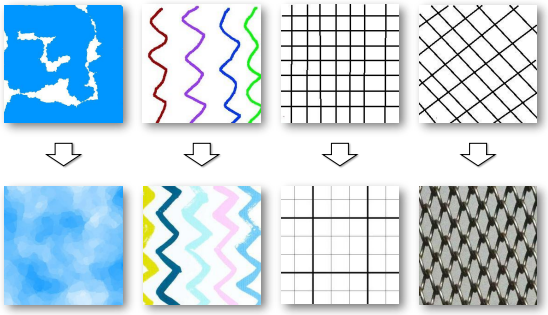

Figure 8: *Examples of drawn sketches (top) used for retrieving a similar texture in the database (bottom).*

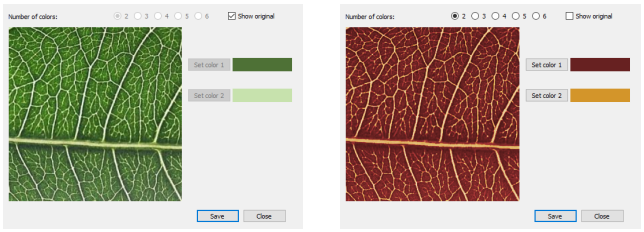

Figure 9: *The image recoloring tool showing the original image (left) and a different choice of coloring using red tones (right).*

where the number of images is very large, and thus were evaluated separately in the second part of the study.

In the evaluation, a Windows 10 system with an Intel i7-8700 CPU, 16GB RAM, and an NVIDIA GeForce GTX 1080Ti GPU with 11GB of VRAM was used. The GUI was displayed at a resolution of 1920×1200. The system was implemented in C++ with machine-learning components running in Python.

In total, 16 users participated in the evaluation. They had no prior experience with our interface but were Computer-Science students. Additional details regarding the evaluation are given in the supplementary material.

5.1. Comparison with alternative methods

Grid view For the first task, the users were asked to retrieve four given textures from the same collection (DTD, containing 1518 texture images), using two different systems. One was a grid view, which consisted of a standard Windows 10 file explorer dialog with large thumbnails, where images were randomly ordered. The second was our interface, where we asked the users to search the textures starting from the overview panel and were free to use all the features our tool provides. Fig. 10 shows the images for this task.

As an introduction to our system, the users were shown a $\sim$1 minute video that illustrated how our interface works (available as supplementary material). The users were not given time to familiarize themselves with the tool. Instead, their first search task was their first interaction with our tool.

Results (Fig. 11) show that the average time to retrieve the target

Figure 10: *Retrieval-task Textures for method comparison. One texture was replicated due to our multi-scale method.*

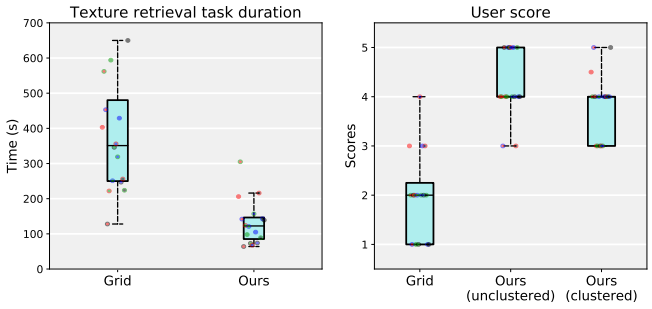

Figure 11: *Retrieval time in seconds (left) and user score (right) for the standard grid view and our texture retrieval system with and without the clustering overlay.*

textures with the grid view (427 seconds) is substantially reduced by using our interface (133 seconds); a more than three-fold improvement even when applied by an inexperienced user.

It is also noteworthy that all participants completed the task faster with our system than with the grid view. Ten users performed faster on each individual search query using our system. The rest performed faster on three out of four search queries using our system but even when just looking at the participants of this second group, the average time for completing the task using the grid view (496 seconds) was still significantly slower than using our system (159 seconds). Furthermore, one of the users failed to retrieve the final target texture via the grid view after searching for more than 17 minutes (only the time until they desisted was accounted in our averages), while this user succeeded via our method in less than three minutes.

With statistical hypothesis testing, retrieval time with our interface is shorter than with the grid view a p-value lower than 0.00001, indicating high statistical significance, and shorter by 4 minutes with a p-value of 0.0024, also statistically significant.

Multi-scale Within this task, we also evaluated the usefulness of our multi-scale replication (Sec. 3.3). One of the textures that we chose as a target for the search (leftmost Fig. 10) was replicated via our scheme because of its multi-scale appearance. In 12 of the 16 cases, the users found the texture via the embedding position corresponding to one of the low-scale (blurred) versions, averaging 27 seconds. The rest obtained the target in its original scale embedding location, averaging 38 seconds. This coincides with our hypothesis that different users first notice features of different scales, and our multi-scale replication takes advantage of this phenomenon to facilitate retrieval in either case.

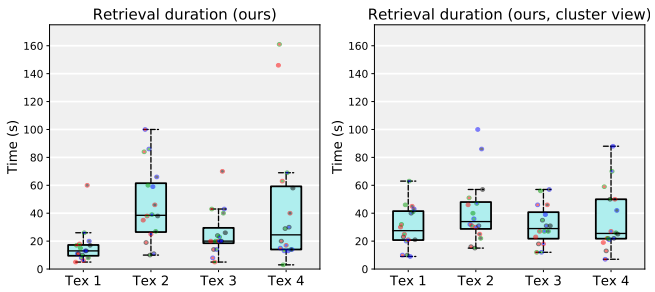

Figure 12: *Individual retrieval time in seconds for the four textures of the initial retrieval (left) and when asked to repeat the task using the cluster view (right).*

Figure 13: *Images retrieved by users using the concept* porous *(top) and* interlaced *(bottom). Only the images marked in red have the corresponding label in the DTD database, showcasing the difference in label interpretation among different users.*

Clustering We asked the participants to repeat the retrieval while enforcing the use of the cluster view of the embedding, to assess the usefulness of the feature. In this case, we randomly rotated the embedding, to prevent users from relying on location knowledge from the previous retrievals. Users were then asked to assess their interaction satisfaction on a Likert scale of 5 for the grid view interface, and our interface with and without using the cluster view. Users rated our interface on average with a 4.25, and the grid view satisfaction with an average of 2.0. (Fig. 11, right). In general, user preference for the *overview* exploration was mixed, as from the 16 participants, 9 preferred the non-clustered view, 4 preferred the clustered view, and 3 gave them the same score.

Fig. 12 gives further insight into the times required to complete the retrieval task with and without requiring the cluster view. In that figure, it can be noted that Textures 2 and 4 required slightly longer search times. For Texture 2, this can be attributed to the fact that it does not exhibit immediately recognizable large-scale features. Indeed, our embedding system groups the texture together with lace-patterned textures, which can only be easily identified when zooming-in. In the case of Texture 4, we believe the dark colors and low contrast of the image may make it less identifiable from a zoomed-out view.

Those users that preferred the non-clustered view commented that the presented clusters did not always meet their expectations, since they grouped textures in an unexpected way. Our clustering implementation is based on the low dimensional 2D position of the textures, but could be improved by taking the high-level feature vector into account as future work to improve the quality of the resulting cluster display.

Text-based retrieval To test keyword use for texture selection, we provided users two out of the 47 labels from DTD: *porous* and *interlaced*. We asked users to retrieve one texture representative of each word, with no time limit. Fig. 13 shows the retrieved textures, with a red highlight for those that have the search keyword as a description label in the DTD database. For the first keyword, *porous*, only half of the participants found a texture that carried the corresponding label in DTD, but all textures exhibit porous features. For the second keyword, *interlaced*, only one of the retrieved textures was labeled as such in the database. Nevertheless, as seen in Fig. 13, all textures feature interlaced patterns. This highlights the difficulty of texture retrieval via labels, as user interpretation of a keyword can vary greatly. In contrast, a texture retrieval system navigated in a semantic way, such as our presented system, provides a more intuitive and general way of approaching such tasks.

5.2. Prioritized t-SNE and image-based search tool evaluation

In this part of the study, we shift our focus to evaluating the importance of the *prioritized-tSNE* and *image-based search* functionality. Since these two features aim at aiding retrieval in large and crowded databases, the texture database used here contains 5824 texture images (see Fig. 14).

The 16 users were randomly divided into two groups, each with 8 users. For group A, users were asked complete a retrieval task with only the overview panel of our interface, and without using the *prioritized-tSNE*, *image-based search*, *multi-scale replication*, or *clustering* functionality. For group B, users were required to start their search with a specified feature, either *prioritized-tSNE* or *image-based search*, for different retrieval targets. After zooming into an area by using one of these features, they continued with that same navigation as group A until finding the target image. The textures for these retrieval tasks are shown in Fig. 15, where the two leftmost ones were used for the *prioritized-tSNE* evaluation and the two rightmost ones for the *image-based search* evaluation. In the latter case, the participants were given a small set of images they could use to start their search. The recorded retrieval time for each texture is shown in Fig. 16.

Prioritized-tSNE In the *prioritized-tSNE* test, the average time of group A (overview mode only) was 354 seconds, whereas group B (*prioritized-tSNE*) took in average 178 seconds, roughly half the time. The results shown in Fig. 16, tasks 1 and 2, indicate that when using our *prioritized-tSNE* the retrieval times were substantially faster for all users except user #13. This user reported being misled by a similar texture (an image of cracked glass), and spent some time exploring the wrong region.

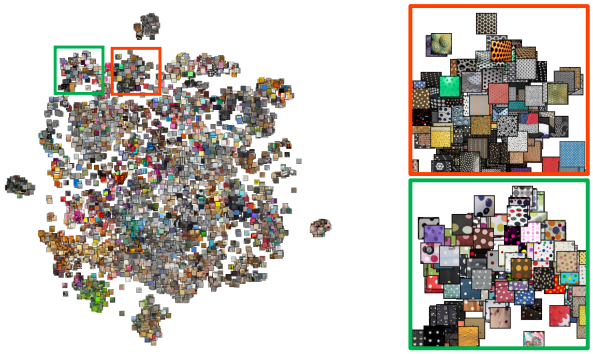

Figure 14: *Embedding of the database of the feature-evaluation study (5824 textures, insets show grouping of similar images).*

Figure 15: *Retrieval-task textures for feature comparison.*

From the feedback in the questionnaire, some users from group A suggested that a tool that can spread the overlapped crowd out would be helpful, which is one of the goals of our *prioritized-tSNE* feature.

Image-based search In the *image-based search* test, group A (*overview* mode only) averaged 650 seconds, and group B (*image search*) averaged 147 seconds, a roughly 4x improvement. In group A, two users (#2 and #4) gave up retrieval after 5 minutes and 3 minutes, respectively. According to their feedback, user #2 switched the retrieval among several possible regions but was finally not able to locate the target. User #4 gave up due to overlapping textures and made a similar suggestion as reported before, that a tool to spread the crowded regions would be useful. In group B, almost all the users succeeded in finishing the task using less time than those in group A, except user #10 for whom the retrieval was stopped at 5 minutes and the reason was similar to that of user #4. This can be probably solved if the users had access to the *prioritized t-SNE* tool. Overall, thanks to the fast identification of the target region when starting with the *image-based search* tool, the retrieval time is substantially shorter.

6. Conclusion

In this paper, we presented a system that facilitates the task of exploration and retrieval in large unlabeled texture collections. We organize images based on semantic features and additionally provide several tools to improve the exploration of image groups that share similar features. One of these tools, our proposed prioritized t-SNE algorithm, can enhance the visualization of areas of interest for a user and might even find applications beyond the scenario of this work. The results of our user study show that our proposed system

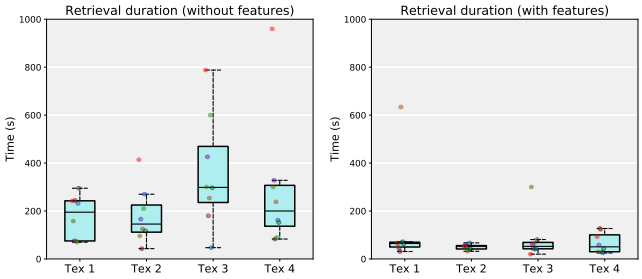

Figure 16: *Individual retrieval time in seconds for the four textures using only the* overview *mode (left) and with additionally the specified features (right). Texture 1 and texture 2 are the retrieval targets for* prioritized-tSNE *evaluation, texture 3 and texture 4 are the retrieval targets for the* image-based search *evaluation.*

is a considerable upgrade to traditional filesystem and grid-based interfaces when exploring texture images.

7. Acknowledgments

The authors would like to thank the University of Oxford for providing the texture database (DTD) used in this paper.

This work is partly supported by VIDI NextView, funded by NWO Vernieuwingsimpuls.

References

[AA16] ANCUTI C., ANCUTI C. O.: Laplacian-guided image decolorization. In *2016 IEEE International Conference on Image Processing (ICIP)* (2016), IEEE, pp. 4107–4111. 5

[AAB11] ANCUTI C. O., ANCUTI C., BEKAERT P.: Enhancing by saliency-guided decolorization. In *CVPR 2011* (2011), IEEE, pp. 257–264. 5

[BEDT08] BEZERRA H., EISEMANN E., DÉCORET X., THOLLOT J.: 3d dynamic grouping for guided stylization. In *Proceedings of the 6th international symposium on Non-photorealistic animation and rendering* (2008), pp. 89–95. 4

[BRL97] BHUSHAN N., RAO A. R., LOHSE G. L.: The texture lexicon: Understanding the categorization of visual texture terms and their relationship to texture images. *Cognitive Science 21*, 2 (1997), 219–246. 1

[BSCL14] BABENKO A., SLESAREV A., CHIGORIN A., LEMPITSKY V.: Neural codes for image retrieval. In *European conference on computer vision* (2014), Springer, pp. 584–599. 2

[CBGM97] CARSON C., BELONGIE S., GREENSPAN H., MALIK J.: Region-based image querying. In *1997 Proceedings IEEE Workshop on Content-Based Access of Image and Video Libraries* (1997), IEEE, pp. 42–49. 2

[CC08] COX M. A., COX T. F.: Multidimensional scaling. In *Handbook of data visualization*. Springer, 2008, pp. 315–347. 3

[CFL*15] CHANG H., FRIED O., LIU Y., DIVERDI S., FINKELSTEIN A.: Palette-based photo recoloring. *ACM Trans. Graph. 34*, 4 (2015), 139–1. 2, 5

[CMK*14] CIMPOI M., MAJI S., KOKKINOS I., MOHAMED S., VEDALDI A.: Describing textures in the wild. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition (2014), pp. 3606–3613. 1, 2, 4, 6

[CRHC18] CHAN D. M., RAO R., HUANG F., CANNY J. F.: t-sne-cuda: Gpu-accelerated t-sne and its applications to modern data. In *2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)* (2018), IEEE, pp. 330–338. 4

[DAEFCO19] DANON D., AVERBUCH-ELOR H., FRIED O., COHEN-OR D.: Unsupervised natural image patch learning. *Computational Visual Media 5*, 3 (2019), 229–237. 2

[DAK10] DINAKARAN B., ANNAPURNA J., KUMAR C. A.: Interactive image retrieval using text and image content. *Cybern Inf Tech 10* (2010), 20–30. 2

[DC14] DONG X., CHANTLER M. J.: Texture similarity estimation using contours. In *BMVC* (2014). 2

[DWF*18] DOWLING M., WENSKOVITCH J., FRY J., HOUSE L., NORTH C.: SIRIUS: Dual, Symmetric, Interactive Dimension Reductions. *IEEE transactions on visualization and computer graphics 25*, 1 (2018), 172–182. 3

[DWL*19] DONG J., WANG L., LIU J., GAO Y., QI L., SUN X.: A procedural texture generation framework based on semantic descriptions. *Knowledge-Based Systems 163* (2019), 898–906. 2

[ENP*09] ELER D. M., NAKAZAKI M. Y., PAULOVICH F. V., SANTOS D. P., ANDERY G. F., OLIVEIRA M. C. F., NETO J. B., MINGHIM R.: Visual analysis of image collections. *The Visual Computer 25*, 10 (2009), 923–937. 3

[FACO17] FRIED O., AVIDAN S., COHEN-OR D.: Patch2vec: Globally consistent image patch representation. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 183–194. 3

[FF10] FEI-FEI L.: Imagenet: crowdsourcing, benchmarking & other cool things. In *CMU VASC Seminar* (2010), vol. 16, pp. 18–25. 3

[FH75] FUKUNAGA K., HOSTETLER L.: The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory 21*, 1 (1975), 32–40. 4

[GAD*20] GUEHL P., ALLEGRE R., DISCHLER J.-M., BENES B., GALIN E.: Semi-procedural textures using point process texture basis functions. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 159–171. 2

[GARL17] GORDO A., ALMAZAN J., REVAUD J., LARLUS D.: End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision 124*, 2 (2017), 237–254. 2

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 2414–2423. 3

[Gib50] GIBSON J. J.: The perception of the visual world. 1

[GMIL08] GOMI A., MIYAZAKI R., ITOH T., LI J.: Cat: A hierarchical image browser using a rectangle packing technique. In *2008 12th International Conference Information Visualisation* (2008), IEEE, pp. 82–87. 3

[GRM*18] GEIRHOS R., RUBISCH P., MICHAELIS C., BETHGE M., WICHMANN F. A., BRENDEL W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231* (2018). 1, 3

[GV19] GARCIA N., VOGIATZIS G.: Learning non-metric visual similarity for image retrieval. *Image and Vision Computing 82* (2019), 18–25. 2

[H*12] HALLEY F., ET AL.: *Perceptually relevant browsing environments for large texture databases*. PhD thesis, Heriot-Watt University, 2012. 3

[HDR19] HU Y., DORSEY J., RUSHMEIER H.: A novel framework for inverse procedural texture modeling. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 1–14. 3

[Hec86] HECKBERT P. S.: Survey of texture mapping. *IEEE computer graphics and applications 6*, 11 (1986), 56–67. 1

[HPvU*16] HÖLLT T., PEZZOTTI N., VAN UNEN V., KONING F., EISEMANN E., LELIEVELDT B., VILANOVA A.: Cytosplore: interactive immune cell phenotyping for large single-cell datasets. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 171–180. 4

[HPvU*17] HÖLLT T., PEZZOTTI N., VAN UNEN V., KONING F., LELIEVELDT B. P., VILANOVA A.: Cyteguide: Visual guidance for hierarchical single-cell analysis. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2017), 739–748. 3

[HVP*19] HÖLLT T., VILANOVA A., PEZZOTTI N., LELIEVELDT B. P., HAUSER H.: Focus+ context exploration of hierarchical embeddings. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 569–579. 3

[Jul62] JULESZ B.: Visual pattern discrimination. *IRE transactions on Information Theory 8*, 2 (1962), 84–92. 2

[Kaz16] KAZAK N.: Performance analysis of spiral neighbourhood topology based local binary patterns in texture recognition. *International Journal of Applied Mathematics Electronics and Computers*, Special Issue-1 (2016), 338–341. 4

[KCB03] KOKARE M., CHATTERJI B., BISWAS P.: Comparison of similarity metrics for texture image retrieval. In *TENCON 2003. Conference on convergent technologies for Asia-Pacific region* (2003), vol. 2, IEEE, pp. 571–575. 2

[KSH12] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105. 2, 3

[LCF*19] LIU L., CHEN J., FIEGUTH P., ZHAO G., CHELLAPPA R., PIETIKÄINEN M.: From bow to cnn: Two decades of texture representation for texture classification. *International Journal of Computer Vision 127*, 1 (2019), 74–109. 2

[Low04] LOWE D. G.: Distinctive image features from scale-invariant keypoints. *International journal of computer vision 60*, 2 (2004), 91–110. 2

[LP15] LUPTON E., PHILLIPS J. C.: *Graphic Design: The New Basics: Revised and Expanded*. Chronicle Books, 2015. 1

[LSS*17] LIU L., SHEN F., SHEN Y., LIU X., SHAO L.: Deep sketch hashing: Fast free-hand sketch-based image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 2862–2871. 2

[LvUH*18] LI N., VAN UNEN V., HÖLLT T., THOMPSON A., VAN BERGEN J., PEZZOTTI N., EISEMANN E., VILANOVA A., CHUVA DE SOUSA LOPES S. M., LELIEVELDT B. P., ET AL.: Mass cytometry reveals innate lymphoid cell differentiation pathways in the human fetal intestine. *Journal of Experimental Medicine 215*, 5 (2018), 1383–1396. 3

[MH08] MAATEN L. V. D., HINTON G.: Visualizing data using t-sne. *Journal of machine learning research 9*, Nov (2008), 2579–2605. 4

[MM96] MANJUNATH B. S., MA W.-Y.: Texture features for browsing and retrieval of image data. *IEEE Transactions on pattern analysis and machine intelligence 18*, 8 (1996), 837–842. 2

[MMB*17] MOHAMED O., MOHAMMED O., BRAHIM A., ET AL.: Content-based image retrieval using convolutional neural networks. In *First International Conference on Real Time Intelligent Systems* (2017), Springer, pp. 463–476. 2

[MWT14] MIZUNO K., WU H. Y., TAKAHASHI S.: Manipulating bilevel feature space for category-aware image exploration. In *2014 IEEE Pacific Visualization Symposium* (2014), IEEE, pp. 217–224. 3

[NAS*17] NOH H., ARAUJO A., SIM J., WEYAND T., HAN B.: Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 3456–3465. 2

[OMS17] OLAH C., MORDVINTSEV A., SCHUBERT L.: Feature visualization. *Distill 2*, 11 (2017), e7. 2, 3

[PFJ03] PERMUTER H., FRANCOS J., JERMYN I. H.: Gaussian mixture models of texture and colour for image database retrieval. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).* (2003), vol. 3, IEEE, pp. III–569. 2

[PG17] PIRAS L., GIACINTO G.: Information fusion in content based image retrieval: A comprehensive overview. *Information Fusion 37* (2017), 50–60. 2

[PHL*16] PEZZOTTI N., HÖLLT T., LELIEVELDT B., EISEMANN E., VILANOVA A.: Hierarchical stochastic neighbor embedding. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 21–30. 3

[PHRC13] PADILLA S., HALLEY F., ROBB D. A., CHANTLER M. J.: Intuitive large image database browsing using perceptual similarity enriched by crowds. In *International Conference on Computer Analysis of Images and Patterns* (2013), Springer, pp. 169–176. 3

[POM07] PAULOVICH F. V., OLIVEIRA M. C. F., MINGHIM R.: The projection explorer: A flexible tool for projection-based multidimensional visualization. In *XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2007)* (2007), IEEE, pp. 27–36. 3

[Por06] PORTA M.: Browsing large collections of images through unconventional visualization techniques. In *Proceedings of the working conference on Advanced visual interfaces* (2006), pp. 440–444. 2, 3

[PS00] PORTILLA J., SIMONCELLI E. P.: A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision 40*, 1 (2000), 49–70. 2

[PTM*20] PEZZOTTI N., THIJSSEN J., MORDVINTSEV A., HÖLLT T., LEW B. V., LELIEVELDT B. P., EISEMANN E., VILANOVA A.: Gpgpu linear complexity t-sne optimization. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of VAST 2019) 26*, 1 (2020). 4

[RBSW99] RODDEN K., BASALAJ W., SINCLAIR D., WOOD K.: Evaluating a visualisation of image similarity as a tool for image browsing. In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis' 99)* (1999), IEEE, pp. 36–43. 3

[Sch16] SCHAEFER G.: Approaches for interactive browsing of large image datasets. In *2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT)* (2016), IEEE, pp. 1–4. 3

[SLJ*15] SZEGEDY C., LIU W., JIA Y., SERMANET P., REED S., ANGUELOV D., ERHAN D., VANHOUCKE V., RABINOVICH A.: Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 1–9. 2

[SS*16] SEETHARAMAN K., SELVARAJ S., ET AL.: Statistical tests of hypothesis based color image retrieval. *Journal of Data Analysis and Information Processing 4*, 02 (2016), 90. 2

[Sub] Substance painter. https://www.substance3d.com/. Accessed: 2020-04-30. 2

[SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014). 2, 3

[Tan19] TAN K. H.: Text-based image retrieval using image captioning. 2

[TT00] TIAN G. Y., TAYLOR D.: Colour image retrieval using virtual reality. In *2000 IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics* (2000), IEEE, pp. 221–225. 3

[WBSS04] WANG Z., BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing 13*, 4 (2004), 600–612. 2

[WK15] WORRING M., KOELMA D. C.: Insight in image collections by multimedia pivot tables. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* (2015), pp. 291–298. 3

[WKZ16] WORRING M., KOELMA D., ZAHÁLKA J.: Multimedia pivot tables for multimedia analytics on image collections. *IEEE Transactions on Multimedia 18*, 11 (2016), 2217–2227. 3

[WL00] WEI L.-Y., LEVOY M.: Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), pp. 479–488. 2

[XCZ*18] XIE X., CAI X., ZHOU J., CAO N., WU Y.: A semantic-based method for visualizing large image collections. *IEEE transactions on visualization and computer graphics 25*, 7 (2018), 2362–2377. 3

[XHZT15] XIE L., HONG R., ZHANG B., TIAN Q.: Image classification and retrieval are one. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* (2015), pp. 3–10. 2

[XZD18] XUE J., ZHANG H., DANA K.: Deep texture manifold for ground terrain recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 558–567. 3

[Yel93] YELLOTT J. I.: Implications of triple correlation uniqueness for texture statistics and the julesz conjecture. *JOSA A 10*, 5 (1993), 777–793. 2

[YFH*06] YANG J., FAN J., HUBBALL D., GAO Y., LUO H., RIBARSKY W., WARD M.: Semantic image browser: Bridging information visualization with automated intelligent image analysis. In *2006 IEEE Symposium On Visual Analytics Science And Technology* (2006), IEEE, pp. 191–198. 3

[YHNYD15] YUE-HEI NG J., YANG F., DAVIS L. S.: Exploiting local features from deep networks for image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (2015), pp. 53–61. 2

[YLLF11] YUE J., LI Z., LIU L., FU Z.: Content-based image retrieval using color and texture fused features. *Mathematical and Computer Modelling 54*, 3-4 (2011), 1121–1127. 2

[YYLZ18] YANG Z., YUE J., LI Z., ZHU L.: Vegetable image retrieval with fine-tuning vgg model and image hash. *IFAC-PapersOnLine 51*, 17 (2018), 280–285. 3

[ZPN13] ZUJOVIC J., PAPPAS T. N., NEUHOFF D. L.: Structural texture similarity metrics for image analysis and retrieval. *IEEE Transactions on Image Processing 22*, 7 (2013), 2545–2558. 2

[ZRPN08] ZHAO X., REYES M. G., PAPPAS T. N., NEUHOFF D. L.: Structural texture similarity metrics for retrieval applications. In *2008 15th IEEE International Conference on Image Processing* (2008), IEEE, pp. 1196–1199. 2

[ZYT17] ZHENG L., YANG Y., TIAN Q.: Sift meets cnn: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence 40*, 5 (2017), 1224–1244. 2