



Texture Retrieval in the Wild Through Detection-Based Attributes

Christian Joppi¹(✉), Marco Godi¹, Andrea Giachetti¹, Fabio Pellacini²,
and Marco Cristani¹

¹ University of Verona, Verona, Italy
christian.joppi@univr.it

² Sapienza University of Rome, Rome, Italy

Abstract. Capturing the essence of a textile image in a robust way is important to retrieve it in a large repository, especially if it has been acquired in the wild (by taking a photo of the textile of interest). In this paper we show that a texel-based representation fits well with this task. In particular, we refer to Texel-Att, a recent texel-based descriptor which has shown to capture fine grained variations of a texture, for retrieval purposes. After a brief explanation of Texel-Att, we will show in our experiments that this descriptor is robust to distortions resulting from acquisitions in the wild by setting up an experiment in which textures from the *ElBa* (an Element-Based texture dataset) are artificially distorted and then used to retrieve the original image. We compare our approach with existing descriptors using a simple ranking framework based on distance functions. Results show that even under extreme conditions (such as a down-sampling with a factor of 10), we perform better than alternative approaches.

Keywords: Texture descriptor · Attribute-based descriptor · Content based image retrieval

1 Introduction

Texels [1] are nameable elements that, distributed according to statistical models (see Fig. 1a–b), form textures that can be defined as *Element-based* [11, 18, 20, 21]. Textures of this kind are of interest in the textile, fashion and interior design industry, since websites or catalogues (containing many products) have to be browsed by users that want to buy or take inspiration from [14, 15]. Two examples taken from the popular e-commerce website Zalando are shown in Fig. 1b. For each item multiple pictures are usually available, including close-up pictures of the fabric highlighting the texture. Not all textures can be defined as Element-based; some can only be characterized at a *micro* scale (*e.g.* in the case of material

C. Joppi, M. Godi—These authors contributed equally to this work.

textures in Fig. 1c), but usually the patterns that decorate textile materials are based on repeated elements.

In the fashion domain browsing for textures is a common task. A shopper that is in possession of an item (e.g. a shirt) with a specific pattern could wish to shop for another item (e.g. pants with a matching pattern) to combine with by taking a close-up picture to highlight the desired texture. A fashion designer could want to take inspiration from an existing garment with only a low resolution picture of the texture available. In these scenarios, it would be useful to be able to search in a database for the desired texture using only a low-quality picture (i.e. in diverse lighting conditions and resolution) as a query. Texture retrieval that is robust to these conditions is an important addition for a fashion e-shop [13, 30] or for fashion designer tools [19]. To be able to achieve this for textures, it is very important to describe them and their structural information in an intuitive and interpretable way, in order to achieve a precise description that enables an accurate retrieval [25] based on the image content.



Fig. 1. (a) Examples of element-based textures in the DTD [5]: the *dotted* (left) and *banded* (right) classes are examples where texels are dots and bands, respectively; (b) Zalando shows for each clothing a particular on the texture; (c) examples of DTD [5] textures which are *not* element-based: (*marbled* on top and *porous* on bottom); here is hard to find clearly nameable local entities; (d) examples of *ElBa* textures: polygon on top, multi-class lined+circle texture on bottom.

For the purpose of achieving a discriminative and nameable description, attribute-based texture features [4, 5, 16, 23] are explicitly suited. In the literature, the 47 perceptually-driven attributes such as *dotted*, *woven*, *lined*, etc. learned on the Describable Texture Dataset (DTD) [5] are the most known.

These 47 attributes are limited in the sense that they describe the properties of a texture image *as a single whole atomic entity*: in Fig. 1a, two different (element-based) attributes are considered: *dotted* (left) and *banded* (right) each one arranged in a column. Images in the same column, despite having the same attribute, are strongly different: for the dots, the difference is on the area; for the bands, the difference is on the thickness. In Fig. 1b (Zalando examples), both garments come with the same “checked” attribute, despite the different sized squares.

It is evident that one needs to focus on the recognizable *texels* that form textures to achieve a finer expressivity.

In this paper, we employ *Texel-Att* [8], a fine-grained, attribute-based texture representation and classification framework for element-based textures.

The pipeline of *Texel-Att* first detects the single texels and describes them by using *individual attributes*. Then, depending on the individual attributes, they are grouped and these groups of texels are described by *layout attributes*.

The *Texel-Att* description of the texture is formed by joining the individual and layout attributes, so that they can be used for classification and retrieval. The dimensionality of the *Texel-Att* descriptor isn't pre-defined, it depends on which attributes are selected for the task. In this paper, we just give some examples to illustrate the general framework.

A Mask-RCNN [10] is used to detect texels; this shows that current state-of-the-art detection architectures can produce element-based descriptions (further improvements are foreseeable as we will discuss later). We design *ElBa*, the first *Element-Based* texture dataset, inspired by printing services and online catalogues¹. By varying in a *continuous* way element shapes and colors and their distribution, we generate realistic renderings of 30K texture images in a procedural way using a total of 3M localized texels. Layout attributes such as local symmetry, stationarity and density are known by construction.

In the experiments we show that, using the attribute-based descriptor that we extract with our framework, we are able to retrieve textures in a more accurate way under simulated image conditions mimicking real-world scenarios. The performance of our approach is compared against state of the art texture descriptors of different kinds to show the usefulness of our approach.

We also show qualitative results to highlight the steps of the employed framework, such as the texel detection (detailed in Sect. 2.1).

2 Method

In this section we explain the *Texel-Att* framework step-by-step. Then we propose a simple method for texture retrieval that can be employed with this framework.

2.1 The *Texel-Att* Framework

Figure 2 shows a block diagram of the *Texel-Att* description creation pipeline.

The main concept is extracting texels using an object detection framework (trained for the task). Then, texels are described with *individual* attributes, *i.e.* labelled according to category, appearance and size. Texels are then grouped and filtered according to the individual labels. For each group, descriptions of the spatial layout of groups are estimated and aggregated into *layout* attributes. The

¹ <https://www.spoonflower.com/>, <https://designyourfabric.ca/>, <https://patternizer.com/d0Wp> and <https://www.contrado.com/> respectively.

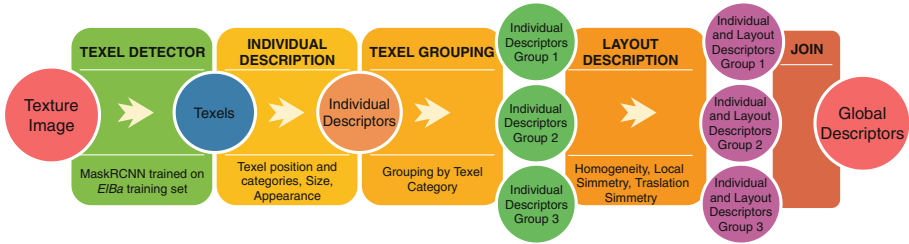


Fig. 2. Block diagram of the formation of the Texel-Att element-based texture descriptor. On the bottom of each plate, the specific choices made in this paper, which can be varied.

composite Texel-Att descriptor is formed by individual and layout attributes. In the following, each processing block is detailed.

Texel Detector. The Mask-RCNN [10] model handles the texel detection by localizing (with bounding boxes and segmentation masks) and classifying objects. The model is trained on the *ElBa* dataset’s training set, learning to detect and classify texels such as *lines*, *circles*, *polygons* (see Sect. 3). Texels are easily handled in any displacement (while a few years ago it was a quite complicated task limited to specific scenarios *i.e.*, lattices [9, 17]).

Individual Description of Texels. By using attributes related to shape and human perception it is possible to characterize each detected texel; in particular we make use of: (i) the *label* indicating its shape, classified by the Mask-RCNN model; (ii) histogram of 11 *colors* using a color naming procedure [28]; (iii) *orientation* of texels; (iv) *size* of texels, represented by the area in pixels. By aggregating (e.g. through averages or histograms, see in the following sections) it is possible to characterize the whole texture. It is worth noting that in this work we are not showing “the best” set of features, but we are highlighting the portability and effectiveness of the framework; in fact, different attributes could be used instead.

Texel Grouping. Texels with the same appearance are clustered, so that spatial characteristics of similar elements can be captured using layout attributes. In this work we simply group texels by the assigned shape labels (*circle*, *line* or *polygon*). Groups with less than 10 texels are removed.

Layout Description of Texels. Spatial characteristics of each texel group, are described by measuring attributes using the spatial distribution of the centroids of the texels. We can refer to the literature on spatial points pattern analysis, where measures for symmetry, randomness, and regularity [2, 7, 29] are available; we select a simple and general set of measures. They are: (i) texel *density*, e.g. the average number of texels per unit of area (for circles and polygons) or line density (e.g. by projecting centroid on to the direction perpendicular to their principal orientation density is measured on one spatial dimension). (ii) Quadratic counts-based *homogeneity* evaluation [12]: the original image is divided into a number

of patches and a χ^2 test is performed to evaluate the hypothesis of average point density in each patch. Similarly to the previous case, we estimated a similar 1D feature on the projection for lines. (iii) Point pair statistics [31]: the histogram of *vectors orientation* is estimated using point pair vectors for all the texel centers. (iv) *Local symmetry*: we considered the centroids' grid for circles and polygons and measured, for 4-points neighborhoods of points, the average reflective self-similarity after their reflection around the central point. The average point distance is used as a distance function. Neighborhood size is used to normalize it. *Translational symmetry* is estimated in a similar way by considering 4-point neighborhoods of the centroids translated by the vectors defined by point pairs in the neighborhood and measuring the average minimum distance of those points. For line texels, we compute on 1D projections.

We report the dimensionalities for each of these attributes in Table 1. Multi-dimensional attributes are histograms, while 1-dimensional ones are averages. By concatenating and Z-normalizing spatial pattern attributes, individual texel attributes statistics and the color attributes of the *background*, the final descriptor for the texture is built.

Table 1. Dimensionality of descriptor attributes. On the left, the attributes computed from the individual characterization of texels; on the right, attributes computed from statistics resulting from the spatial layout. The total dimensionality of the descriptor is 36.

Label Histogram	Color Histogram	Orientation Histogram	Size	Total	Density	Homogeneity	Vector Orientations	Local Symmetry	Traslational Symmetry	Background Color	Total
3	11	3	1	18	1	1	3	1	1	11	18

2.2 Element-Based Texture Retrieval

The descriptor detailed in the previous section can be used to compute distances between element-based textures using the corresponding attributes. We define *database set* the set of images that we want to search into using a *query image*. The idea is that database texture closest to the query image (in terms of descriptor distance) are also the most similar ones in the database set.

The pipeline is as follows: a query image (e.g. a picture of a textured captured by a user) is processed by the Texel detector, allowing for the computation of individual and layout attributes and thus obtaining a descriptor. A standard distance function (such as cosine distance) is computed between every database image and the query image. The database set is then sorted according to the distance and the resulting ranking can be shown to the user for browsing.

3 *ElBa*: Element-Based Texture Dataset

While available datasets such as the DTD [5] include some examples of element-based textures mixed with other texture types (Fig. 1(a)), there is no dataset

focused on this particular domain. In this work, we present *ElBa*, the first element-based texture dataset. As shown in Fig. 1(d), photo-realistic images are included in the *ElBa* dataset. Training a model with synthetic data is a common practice [3,27] and annotations for texels are easily made available as an output of the image generation process. *Layout* attributes and *individual* ones (addressing the single texel) can be varied in our proposed parametric synthesis model. For example individual attributes such as texel shape, size and orientation and color can be varied. Available shapes are *polygons* (squares, triangles, rectangles), *lines* and *circles* (inspired by the 2D shape ontology of [24]). The idea is that these kind of shapes are common in geometric textiles and they approximate other more complex shapes. Orientation and size are varied within a range of values. We choose colors from color palettes to simulate a real-world use of colors.

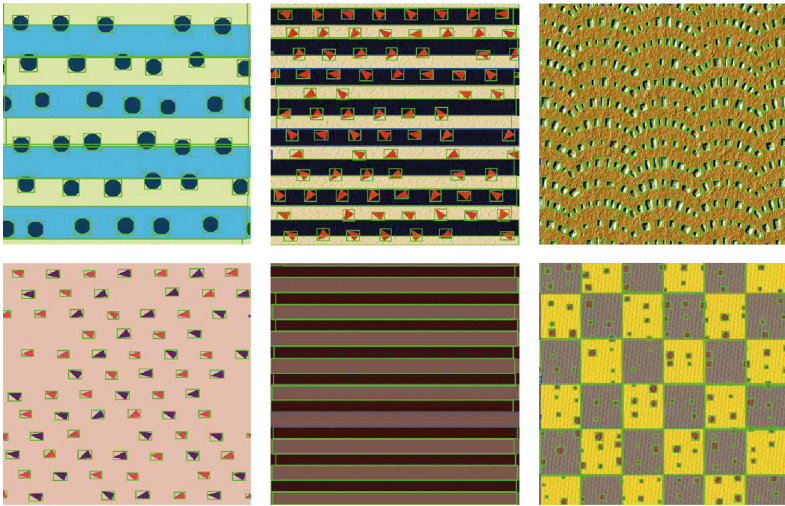


Fig. 3. Texel-Att detection qualitative results on *ElBa* datasets. In green the correct detections, in red the false positives and in blue the false negatives. (Color figure online)

As for Layout Attributes, we select different 2D layouts based on symmetries to place texels. Linear and grid-based layouts are considered; one or two non-orthonormal vectors define the translation between texels in the plane. With this parametrization, we can represent several tilings of the plane. As for randomized distributions, we jitter the regular grid, creating a continuous distribution between randomized and regular layouts.

We also consider multiple element shapes within a single image, creating for example dotted+striped patterns. Each group of elements of the same shape is distributed with its own spatial layout, creating arbitrary multi-class element textures as in Fig. 1(d).

We made use of Substance Designer for pattern generation (which gives high-quality output and pattern synthesis, and is easily controllable) and IRay (which is a physically-based renderer)². Substance gives high-quality pattern synthesis, easy control and high-quality output including pattern antialiasing. Low-frequency distortions of the surface of the plane where the pattern is represented and high frequency patterns are added to simulate realistic materials.

A total of 30K texture images (for a total 3M annotated texels) rendered at a resolution of 1024×1024 has been generated by this procedure. For each image ground-truth data (such as texel masks, texel bounding boxes and attributes) is available. *ElBa* does not come with a partition into classes: differently from other datasets used in texture analysis semantic labels for classification tasks can be computed from ground truth attributes or by user studies.

The dataset is randomly partitioned with a 90/10 split for, respectively, training and testing set.

Table 2. *AUC (Area Under Curve)* for each distortion variant. Texel-Att performs better on every one of them. The related CMC are shown in Fig. 4.

Distortions	Tamura [26]	FV-CNN [6]	Texel-Att
Down-sampling (100×100) and impulsive noise ($p = 0.2$)	0.1380	0.3304	0.6618
Down-sampling (200×200) and impulsive noise ($p = 0.2$)	0.2103	0.4811	0.8011
Down-sampling (300×300) and impulsive noise ($p = 0.2$)	0.2284	0.5640	0.8560
Down-sampling (100×100) and radial lighting effect	0.1611	0.4394	0.6356
Down-sampling (200×200) and radial lighting effect	0.1728	0.8001	0.8746
Down-sampling (300×300) and radial lighting effect	0.2708	0.8855	0.9376

4 Experiments

Experiments show the potential of our framework for the description of element-based textures, with a focus on difficult environmental conditions (low resolution and diverse lighting) ensuring an accurate retrieval inside large catalogues of textures in real-world applications.

² <https://www.allegorithmic.com/> and <https://bit.ly/2Hz4ZVI> respectively.

4.1 Qualitative Detection Results

We briefly show the detection results over our dataset, a fundamental step of our framework, through some qualitative results in Fig. 3. Texels are highlighted by bounding boxes which are then used to compute the attributes (described in Sect. 2.1) that we employ in the following experiment.

4.2 Texture Retrieval Results

In this experiment, we highlight the effectiveness of Texel-Att in a retrieval task under simulated real-world conditions following the procedure detailed in Sect. 2.2. We compare our approach with both state-of-the-art texture descriptor FV-CNN [5] and Tamura attribute-based descriptor [26]. The *database set* for this retrieval experiment is the whole test partition of the *ElBa* dataset (composed of ~ 3000 images). To simulate the real challenging conditions, we generated 6 variants of each image, down-sampling at one of 3 different resolutions (100×100 , 200×200 , 300×300) and up-sampling them back to the original image size (1024×1024). Then we apply one of the following distortions:

- impulsive noise with a pixel’s probability of 0.2 over all the image;

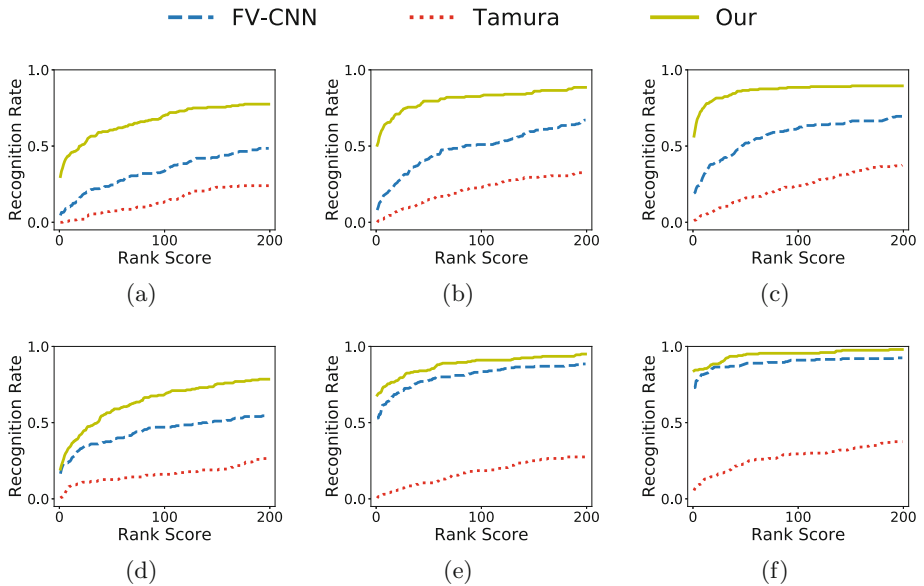


Fig. 4. CMC curves on the retrieval experiments. Different plot for different variants of distortion: (a) 100×100 down-sampling and impulsive noise (b) 200×200 down-sampling and impulsive noise (c) 300×300 down-sampling and impulsive noise (d) 100×100 down-sampling and radial lighting effect. (e) 200×200 down-sampling and radial lighting effect. (f) 300×300 down-sampling and radial lighting effect. On the x axis the rank score (first 200 positions). On the y axis the recognition rate.

- radial lighting effect, increasing the brightness on a random point on the image and gradually decreasing it more in each pixel the farther from the chosen point it is.

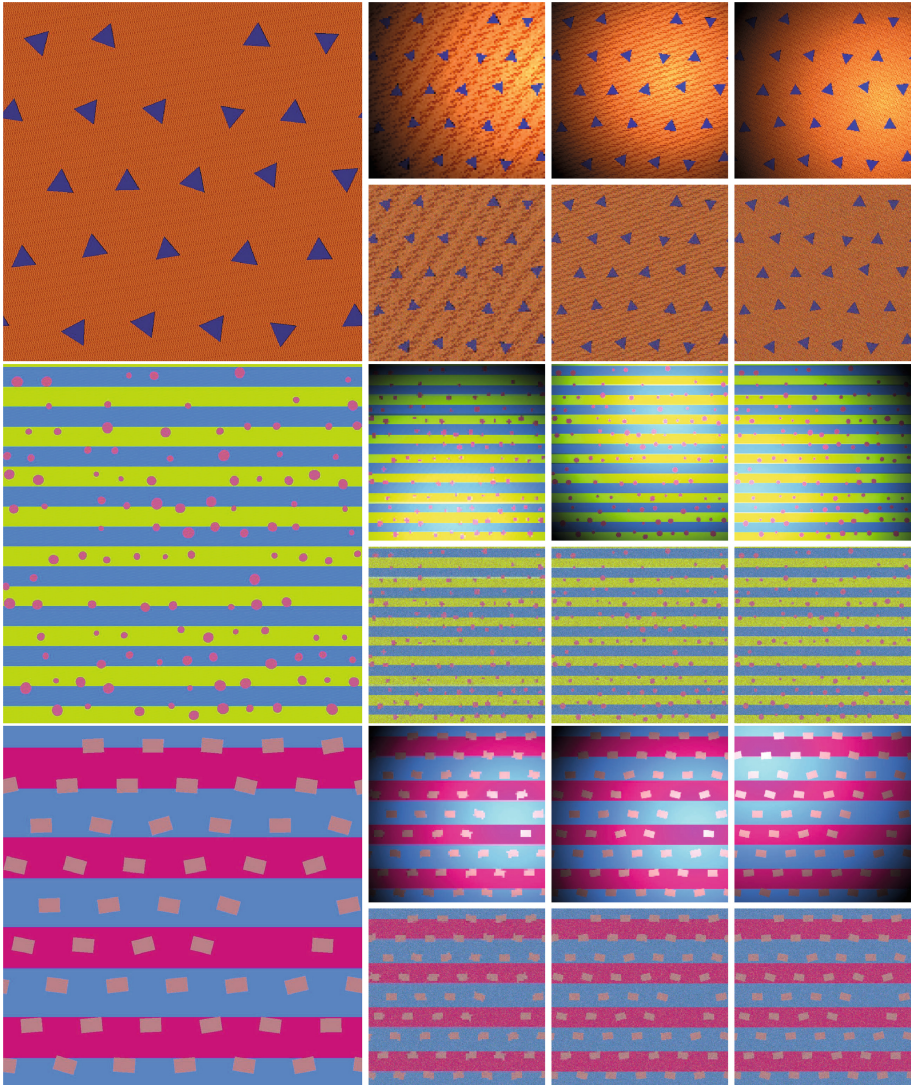


Fig. 5. Three examples of distortions. For each one the biggest image is the original pattern. On the right, the first row depicts the radial lighting effect while the second one the impulsive noise distortion. The column are organized from the 100×100 down-sampling to 300×300 down-sampling.

Some examples of these images are shown in Fig. 5. It can be seen that distorted images simulate pictures that could be captured by users wishing to employ a retrieval application. The lighting effect simulates the flash of a camera while impulsive noise simulates general defects in the image acquisition process.

We consider each of the 6 variants as *query set* and we test each one separately. Given a distorted image from the query set, the task is to retrieve the corresponding original one from the database set. The position of the correct match in the computed ranking is recorded. This process is repeated for every image in a query set.

To distance functions used for ranking is chosen according to the descriptor; for each descriptor we selected the best performing distance function between all of the ones available in the MATLAB software [22]. More specifically, for the FV-CNN descriptor and our descriptor we employ the cosine distance while for the Tamura descriptor the cityblock distance function performs best.

Table 2 shows the results of this experiment in all of the 6 variants previously described. In each case Texel-Att reaches the best results in terms of *AUC: Area Under Curve* index related to CMC (Cumulative Matching Characteristics) curves shown in the plots in Fig. 4. We show only the first 200 positions for the CMC curve rank as we consider higher ranking positions less useful for a retrieval application (a user will rarely check results beyond 200 images).

5 Conclusion

This paper promotes to describe element-based textures by using attributes which focus on texels. Our framework, Texel-Att, can successfully describe and retrieve this type of patterns inside large databases even under simulated real-world factors such as poor resolution, noise and lighting conditions. The experiments show that we perform better in this task with our texel based attributes than by using state-of-the-art general texture descriptors, paving the way for retrieval applications in the fashion and textile domains where element-based textures are prominent.

Acknowledgements. This work has been partially supported by the project of the Italian Ministry of Education, Universities and Research (MIUR) “Dipartimenti di Eccellenza 2018–2022”, and has been partially supported by the POR FESR 2014–2020 Work Program (Action 1.1.4, project No. 10066183). We also thank Nicolò Lanza for assistance with Substance Designer software.

References

1. Ahuja, N., Todorovic, S.: Extracting texels in 2.1 D natural textures. In: 2007 IEEE 11th International Conference on Computer Vision, pp. 1–8. IEEE (2007)
2. Baddeley, A., Rubak, E., Turner, R.: Spatial Point Patterns: Methodology and Applications with R. Chapman and Hall/CRC, New York (2015)

3. Barbosa, I.B., Cristani, M., Caputo, B., Rognhaugen, A., Theoharis, T.: Looking beyond appearances: synthetic training data for deep CNNs in re-identification. *Comput. Vis. Image Underst.* **167**, 50–62 (2018)
4. Bormann, R., Esslinger, D., Hundsdoerfer, D., Haegele, M., Vincze, M.: Robotics domain attributes database (RDAD) (2016)
5. 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*, pp. 3606–3613 (2014)
6. Cimpoi, M., Maji, S., Kokkinos, I., Vedaldi, A.: Deep filter banks for texture recognition, description, and segmentation. *Int. J. Comput. Vis.* **118**(1), 65–94 (2016)
7. Diggle, P.J., et al.: *Statistical Analysis of Spatial Point Patterns*. Academic press, London (1983)
8. Godi, M., Joppi, C., Giachetti, A., Pellacini, F., Cristani, M.: Texel-Att: representing and classifying element-based textures by attributes (2019)
9. Gui, Y., Chen, M., Ma, L., Chen, Z.: Texel based regular and near-regular texture characterization. In: *2011 International Conference on Multimedia and Signal Processing*, vol. 1, pp. 266–270. IEEE (2011)
10. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988. IEEE (2017)
11. Ijiri, T., Mech, R., Igarashi, T., Miller, G.: An example-based procedural system for element arrangement. In: *Computer Graphics Forum*, vol. 27, pp. 429–436. Wiley Online Library (2008)
12. Illian, J., Penttinen, A., Stoyan, H., Stoyan, D.: *Statistical Analysis and Modelling of Spatial Point Patterns*, vol. 70. Wiley, Hoboken (2008)
13. Jing, Y., et al.: Visual search at pinterest. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1889–1898. ACM (2015)
14. Kovashka, A., Parikh, D., Grauman, K.: WhittleSearch: image search with relative attribute feedback. In: *2012 IEEE CVPR*. IEEE (2012)
15. Kovashka, A., Parikh, D., Grauman, K.: WhittleSearch: interactive image search with relative attribute feedback. *Int. J. Comput. Vis.* **115**(2), 185–210 (2015)
16. Liu, L., Chen, J., Fieguth, P.W., Zhao, G., Chellappa, R., Pietikäinen, M.: A survey of recent advances in texture representation. *CoRR abs/1801.10324* (2018). <http://arxiv.org/abs/1801.10324>
17. Liu, S., Ng, T.T., Sunkavalli, K., Do, M.N., Shechtman, E., Carr, N.: PatchMatch-based automatic lattice detection for near-regular textures. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 181–189 (2015)
18. Loi, H., Hurtut, T., Vergne, R., Thollot, J.: Programmable 2D arrangements for element texture design. *ACM Trans. Graph.* **36**(4) (2017). <https://doi.org/10.1145/3072959.2983617>
19. Hadi Kiapour, M., Han, X., Lazebnik, S., Berg, A.C., Berg, T.L.: Where to buy it: matching street clothing photos in online shops. In: *International Conference on Computer Vision* (2015)
20. Ma, C., Wei, L.Y., Lefebvre, S., Tong, X.: Dynamic element textures. *ACM Trans. Graph.* **32**(4), 90:1–90:10 (2013). <https://doi.org/10.1145/2461912.2461921>
21. Ma, C., Wei, L.Y., Tong, X.: Discrete element textures. *ACM Trans. Graph.* **30**(4), 62:1–62:10 (2011). <https://doi.org/10.1145/2010324.1964957>
22. MATLAB: version R2019a. The MathWorks Inc., Natick, Massachusetts (2019)
23. Matthews, T., Nixon, M.S., Niranjan, M.: Enriching texture analysis with semantic data. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1248–1255 (2013)

24. Niknam, M., Kemke, C.: Modeling shapes and graphics concepts in an ontology. In: SHAPES (2011)
25. Smeulders, A.W., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(12), 1349–1380 (2000)
26. Tamura, H., Mori, S., Yamawaki, T.: Textural features corresponding to visual perception. *IEEE Trans. Syst. Man Cybern.* **8**(6), 460–473 (1978)
27. Tremblay, J., et al.: Training deep networks with synthetic data: bridging the reality gap by domain randomization. In: *Proceedings of CVPR Workshops* (2018)
28. Van De Weijer, J., Schmid, C., Verbeek, J., Larlus, D.: Learning color names for real-world applications. *IEEE Trans. Image Process.* **18**(7), 1512–1523 (2009)
29. Velázquez, E., Martínez, I., Getzin, S., Moloney, K.A., Wiegand, T.: An evaluation of the state of spatial point pattern analysis in ecology. *Ecography* **39**(11), 1042–1055 (2016)
30. Yang, F., et al.: Visual search at eBay. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2101–2110. ACM (2017)
31. Zhao, P., Quan, L.: Translation symmetry detection in a fronto-parallel view. In: *CVPR 2011*, pp. 1009–1016. IEEE (2011)