Texture Similarity Estimation Using Contours

Xinghui Dong
http://www.macs.hw.ac.uk/texturelab

Mike J. Chantler
http://www.macs.hw.ac.uk/~mjc/

The Texture Lab
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh, UK

Abstract

In a study of 51 computational features sets Dong et *al*. [1] showed that none of these managed to estimate texture similarity well and, coincidentally, none of these computed higher order statistics (HOS) over large regions (that is larger than 19×19 pixels). Yet it is well-known that the human visual system is extremely adept at extracting long-range aperiodic (and periodic) "contour" characteristics from images [5, 6]. It is our hypothesis that HOS computed over larger spatial extent in the form of contour data are important for estimating perceptual texture similarity. However, to the authors' knowledge the use of contour data (rather than edge data) has not been proposed before as the basis for a set of feature vectors.

We provide results of an experiment with 334 textures that shows that contour data is more important than local image patches, or 2nd-order global data, to human observers.

We also propose a contour-based feature set that exploits the long-range HOS encoded in the spatial distribution and orientation of contour segments. We compare it against the 51 feature sets tested by Dong et *al*. [1, 2] and another contour model derived from shape recognition. The results show that the proposed method outperforms all the other feature sets in a pairs-of-pairs task and all but two feature sets in a ranking task. We attribute this promising performance to the fact that this new feature set encodes long-range HOS.

1 Introduction

Although performances in the high nineties are typically obtained for tasks such as texture segmentation and classification the same cannot be said of judging texture similarity where a classifier has to estimate the *degree* to which pairs of textures appear similar to human observers[1]. In an investigation of 51 feature sets Dong et *al*. [1] showed that none of these managed to estimate similarity data derived from a population of human observers better than an average agreement rate of 57.76%. Coincidentally, none of these computed higher order statistics (HOS) over large regions (≥ 19×19 pixels).

[1] Such perceptual similarity data are useful for a variety of tasks, from measuring the perceived difference between the appearance of textures (e.g. the visual difference between a worn carpet and a new sample) to simply ranking the results of search results.

While it is generally accepted that visual texture can be represented by spatial statistics and despite over thirty years' of texture research there is still little agreement as to the type, order or spatial extent over which these should be calculated.

First order statistics are by nature computed without reference to the spatial arrangement of pixels and are little used in texture analysis. Second order statistics such as those computed by the autocorrelation function encode information concerning periodicities and are often derived by applying nonlinear functions (variance estimators) to bandpass (linear) filters [3]. They are computed at a wide variety of spatial extents depending upon the task and the size of the region concerned. In contrast higher order statistics are often computationally intensive to derive, and therefore computed over limited spatial extent[2]; "texton" and other vector quantization methods typically being limited to 19×19 pixel neighbourhoods [1, 2].

Thus, the information that texture features are derived from tends to fall into two categories: (1) second order periodic data computed over a variety of scales and (2) short-range aperiodic information. We have discovered few methods that encode long-range, aperiodic characteristics of texture; however, it is well-known that such data are critical to human perception of imagery [5-8]. For instance, scrambling phase spectra (while leaving the power spectra intact) will often render imagery unintelligible to the human observer [8]. It is also well-known that humans are extremely adept at exploiting the long-range visual interactions evident in contour information [5, 6]. However, to the authors' knowledge no research has been reported that exploits contour information for texture analysis.

Our key hypothesis is therefore that "contour" information is important to perceptual texture analysis and in this paper we test this hypothesis in two ways. First, in Section 2, we report the results of an experiment with human observers in order to determine which of three different types of information (2nd-order statistics, local higher order statistics and contour information) are more important for the perception of texture. We go on in Section 3 to develop a new feature set derived from contour segment data and in the penultimate section we test this against the same 51 feature sets that Dong et *al*. [1, 2] used in their study and a shape recognition based feature set. To our knowledge such contour-based texture features that exploit long-range HOS have not been investigated before.

2 The Importance of Three Types of Data to Texture Perception

The key hypothesis of this paper is that contours are important to the human perception of texture and that, in particular, they are more important than the two types of data commonly exploited by today's computational texture features. These two types of data are: (1) 2nd-order statistics encoded in the power spectrum which are typically used by filtering-based features to encode periodicities at both long-range and short-range, and (2) the HOS available from local image patches used in texton-based or other neighbourhood-based features to represent short-range aperiodic spatial relationships. We therefore used three types of stimuli in our experiment: (1) contour maps, (2) power spectrum only images [8] and (3) randomized, blocked images [1]. Samples of each are shown in the lower row of Figure 1.

[2] However, pyramid decompositions [4] can be utilized to enhance the spatial extent that computational features exploit at the cost of blurring the data used at the higher levels in the pyramid.

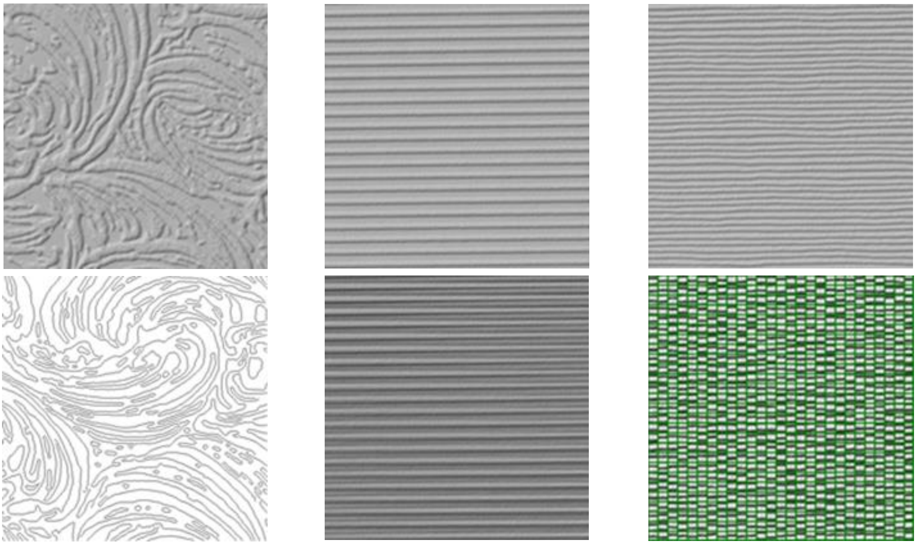

Figure 1: Each of the three columns shows two images derived from the same texture sample (although not the same physical texture area). The upper row shows unprocessed images. The lower row shows, from left to right, the corresponding contour map, power spectrum image and randomized, blocked image.

The contour maps were produced using the Canny edge detector [9]. The power spectrum images were generated by scrambling their phase spectra in the frequency domain [8]. The blocked images used random placement of 19×19 image patches with green borders [1]. (The latter was designed to reduce the effect of newly formed short-range interactions caused by discontinuities between newly neighbouring blocks).

Ten human observers were used in a 2AFC (two-alternative forced choice) scheme with 334 texture images drawn from the *Pertex* database [10]. In each trial the observer was required to compare an original texture image quarter and one variant image quarter ("variant" being one of either contour, power spectrum or randomized block) and decide whether the variant represented the original texture or not (50% of the time they did not). Different quarters of the same texture sample were used in order to prevent observers from performing pixel-wise comparisons. Each of the 10 observers performed 334 trials for each type of variant image. The results are shown in Table 1 below.

Subset	Contour	Power Spectrum	Randomized Blocked	Intersection
Size	247/334	207/334	157/334	92/334

Table 1: The numbers of samples from the 334 texture database that can be recognized using the three different types of variant images.

We were surprised at how useful power spectra were to observers, especially compared to the randomized block images. Inspection of the database revealed many "periodic regular" textures that are well represented by power spectra. The most important point however, is that contour images provided the most relevant information to observers allowing them to correctly recognize 247 out of the 334 texture samples.

3 Computing Spatial Distributions of Contour Segments

This section introduces a new contour-based texture feature set. Essentially, each contour is extracted and encoded as a set of segments. We use these data in two ways as outlined in Figure 2. In the first we encode the average shape of the contours in a segment joint orientation/distance histogram. This provides data on the long-range higher-order visual interactions that these contours provide. In the second we encode the spatial distributions and orientations of the all of the segments within a local window without regard to which contour they belong. These data naturally provide relatively short-range (23×23 or less) HOS.

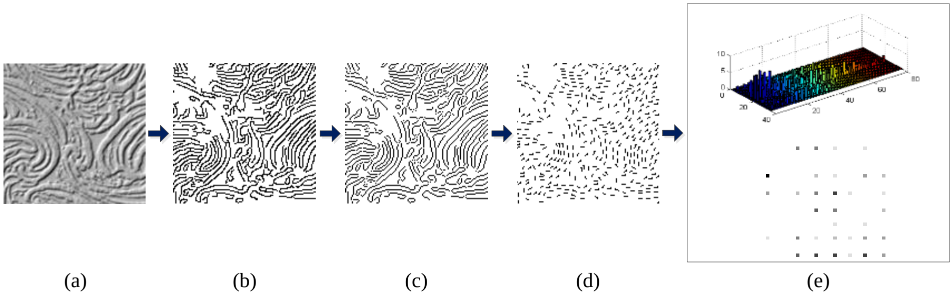
(a) (b) (c) (d) (e)

Figure 2: A representation of the basic information flow: (a) original texture image; (b) edge map; (c) skeleton map; (d) segment map. For display purposes, only a part of pixels are shown for each approximate segment; and (e) the joint histogram (upper) and basic aura matrix [11] (lower, only one is shown here).

In the next section we briefly describe how segment maps together with their underling contour information are extracted. In following section (3.2) we describe how the spatial distribution and orientation information are used to compute the two types of data used within the new feature vectors.

3.1 Producing the Segment Maps (Figure 2 (d))

Although primitives or salient points of contours are commonly utilized for their representation [12], the associated computation is complicated, especially, when large numbers of contours have to be processed. As there has been much research reported on representing objects using fragmented contour segments [7, 12] we were inspired to do likewise. We first fragment a contour into a set of equidistant segments and then encode the spatial distribution and orientation of these segments. Note that images are first processed with the Canny edge detector [9] followed by a morphological erosion operator [13] in order to produce skeleton maps (see Figure 2 (c)).

Given that each contour is regarded as a component, connected component labelling [14] is performed on a skeleton map. Subsequently, the Moore-neighbour tracing algorithm with Jacob's stopping criteria [13] is applied to each contour and a sequence of points is obtained from each contour. However, the exterior boundary of one contour is derived rather than the contour itself because the tracing algorithm considers each contour as a region. The traversing sequence of a contour is further obtained from its exterior boundary sequence. Given that a contour contains a sequence of points: $P_1 \dots P_n$ with coordinates of $(x_1, y_1) \dots (x_n, y_n)$, its length (CL) is computed as below:

$$CL = \sum_{i=1}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}. \quad (1)$$

When the length of the segment is set as SL, the contour is divided into $M = \lfloor CL/SL \rfloor$ segments. In addition, any contour whose length (CL) is smaller than SL will be removed.

Due to the importance of local orientations to the perception of texture structure [15], we represent segments by their mid-point position (on themselves) and chord orientation θ ($\theta \in$ (0°, 180°]). Figure 3 presents three sets of typical segment shapes and their approximate chords. The result is the segment map which encodes each contour as a set of labelled segments, i.e. their mid-points and chord orientations.

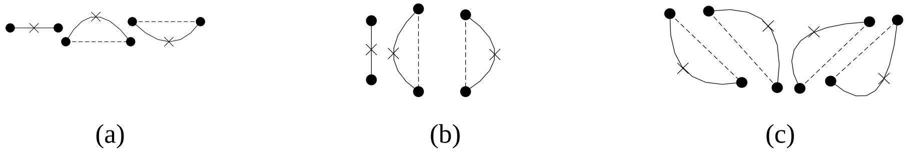

Figure 3: The solid lines above represent example contour segments, the solid dots represent segment endpoints, the dotted lines show the chords of the segments, while the crosses show the segment mid-points. The orientations of the chords and the mid-points are used to represent contours.

3.2 Encoding Contours' Segment Maps

We use two different approaches to represent the spatial distribution and orientation of contours' segments. In the first we compute an average segment distribution across contours (that is we compute pair-wise segment relationships within contours and then average across all contours in an image). In the second we use the basic aura matrix [11] to compute segment co-occurrence data with no regard to which contour they belong. In the latter case we restrict the pairs to those occurring within a local $L \times L$ neighbourhood.

3.2.1 Encoding the Average Shape of Contours within an Image

Since the orientation difference is regarded as the approximation of the local curvature and can provide better discriminatory power, we use this to encode the change of contour direction. In addition, the distance between the mid-points of M segments within a contour is also employed to capture their spatial layout. Pair-wise orientation differences and pair-wise distances (between segments) are computed for all $(M-1)(M-2)/2$ segment pair combinations.

The contour segment joint histogram (which we refer to as "CSJH") of the orientation differences and distances is accumulated. Note that the angle θ was quantized into $A \in \{18, 36\}$ bins, providing two possible histogram resolutions for the evaluation (Section 4.1). It is these histograms that are used to represent individual contours. Also the histograms are averaged across contours to produce a single average contour histogram per image. Of course the final histogram could have been computed without the intermediate step of computing individual contour histograms; however, what is important is that the segment pairs are restricted to those available within single contours.

3.2.2 Representing the Spatial and Angular Distribution of the Segments across Contours

In this feature we compute segment relationships within an image but the mapping of segments to contours is ignored. In this case it is computationally too expensive to compute all pair-wise segment data within an image. Instead we adapt basic aura matrices [11] to compute segment-to-segment angle and position relationships restricted to a local $L \times L$ neighbourhood. Basic aura matrices normally comprise sets of 2D (co-occurrence) histograms where the axes represent the two grey-levels of the pairs of pixels. In our case the axes represent the two angles of the pairs of segments. These angle co-occurrence histograms are generated for different pair sets, where the segment pairs in a pair set are defined by a displacement vector in a similar way to that used for grey level co-occurrence matrices. Thus they represent, for instance, how many pairs of segments exist within an image that are separated by the displacement vector $d = (\Delta x, \Delta y)$ ($|\Delta x|, |\Delta y| \leq \lfloor L/2 \rfloor$, where L is the width of the neighbourhood) and that have angles θ_1 and θ_2. We use the term "basic segment orientation aura matrices" (BSOAMs) to refer to these matrices and their values are used directly in the feature vector. (Note that neighbourhood size was set as $L = 2SL + 1$, where SL is the segment length and $SL \in \{3,5,7,9,11\}$ and therefore the maximum sized neighbourhood considered was 23×23 pixels).

3.2.3 Generating the Contour-based Feature Vector

The mean of all CSJHs and each BSOAM are concatenated into one feature vector which we refer to as "SDoCS" (spatial distribution of contour segments). We test it with two different segment angle quantization schemes (using A bins, $A \in \{18, 36\}$) and five different segment lengths ($SL \in \{3,5,7,9,11\}$) and one multi-scale case ($SL =$ "MS") which concatenates all five feature vectors derived from the five different segment lengths.

4 Experimental Design

Three hundred and thirty-four textures and two different similarity tasks were used to assess the performance of the new contour-based feature set against 52 existing feature sets (51 as investigated by Dong et *al*. [1, 2] and one contour type feature derived from shape recognition: chain code histogram (CCH) [16]).

The first task was a pair-of-pairs application and the second was a conventional ranking problem. In the former the classifier is presented with two pairs of textures and must decide on which pair differs most. Human-derived ground-truth for this task (1000 trials) was available from Clarke et *al*. [17]. For the ranking ground-truth we used a perceptual similarity matrix generated from a free-grouping experiment performed by human observers and processed using Isomap analysis [18] to provide finer-grained distinctions [2, 17]. We call this dataset 8D-ISO because it was found that 8 dimensions were sufficient to encode the majority of the variance in the similarity matrix. Note that it was the availability of these real-valued similarity data that dictated our choice of using the *Pertex* texture database.

The performance of the feature sets was assessed by first using these to compute 334×334 texture similarity matrices and then using these matrices in the pairs-of-pairs and ranking tasks.

4.1 Using the Texture Features to Compute Similarity Matrices

Each 1024×1024 texture image is decomposed into 5 Gaussian pyramid levels [4]. Each level is separately normalized to an average intensity of 0 and standard deviation of 1. Feature vectors were computed at all levels and combined into a single multi-resolution feature vector. In addition, the original resolution (1024×1024) feature vectors were also examined in this study.

The *Chi-square* statistic [19] is used to calculate pair-wise distances for histogram-based and aura matrix-based feature sets, while the *Euclidean* distance is utilized for all other feature sets. These distances are normalized to [0, 1] and subtracted from 1 to provide data for the similarity matrices. These simple distance (similarity) metrics were used as we did not want to encounter problems with overtraining machine learning methods that would inevitably occur with what is a relatively small texture set (but which does have the advantage, unlike other datasets, that it is available with a rich, non-binary similarity matrix which allows complete ranking of retrieval [2]).

4.2 Evaluation Methods

Having obtained the similarity matrices from the computational features it is then a simple task to use these to generate either pairs-of-pairs judgements or retrieval rankings (the latter being generated using a query image taken from the original database). In the case of the former the agreement rate (%) between the computational and the 1000 human pairs-of-pairs' judgements is used as the performance metric [1]. For the ranking-based assessment we compared the rankings of the computational and human derived retrievals (which excluded the query image) using a measure G ($G \in [0, 1]$) proposed by Fagin et *al.* [20]. We did this for the top $N \in$ {10, 20, 40} retrieved textures. This measure has the advantage that it takes into account the relative rankings of the computational and human derived retrievals.

5 Experimental Results

5.1 Pair-of-Pairs Task

Results are shown for two resolutions in Figure 4. For the pair-of-pairs task the best feature set of the 51 feature sets tested in [1] is the Multi-resolution Simultaneous Autoregressive Model (MRSAR) [21]. This is therefore shown separately in Figure 4 together with the average performance of all of these features (as "MeanOf51"). In addition, the results of one shape recognition-based feature set, namely, chain code histogram (CCH), are also reported. The remainder of the graph shows the results for our contour-based feature set at two different segment angle quantization schemes and six different segment lengths ($SL \in \{3,5,7,9,11, MS\}$).

It can be observed that (1) the performance of all feature sets are enhanced when the multi-resolution scheme is used; (2) our feature set performs better when segment angle θ is quantized into 36 angle bins than 18 and with longer segment lengths where it outperforms the best conventional feature set MRSAR.

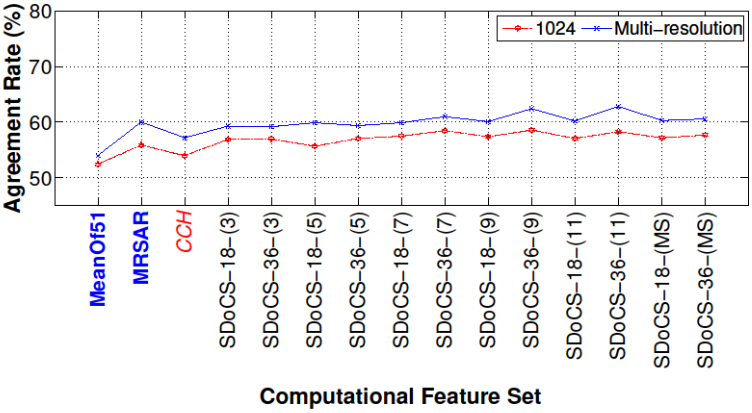

Figure 4: Agreement rates of computational features obtained against human pair-of-pairs data computed at a resolution of 1024×1024 (red trace) and all 5 resolutions combined (blue trace). The first two columns (“MeanOf51” and “MRSAR”) show the mean and best results obtained using the 51 feature sets tested in [1]. The next column shows results obtained using the Chain Code Histogram (CCH). The remaining results labelled in black “SDoCS-A-SL” are results for our new feature set where the segment angle θ is quantized into A bins and the segment lengths SL are taken from $\{3,5,7,9,11,MS\}$.

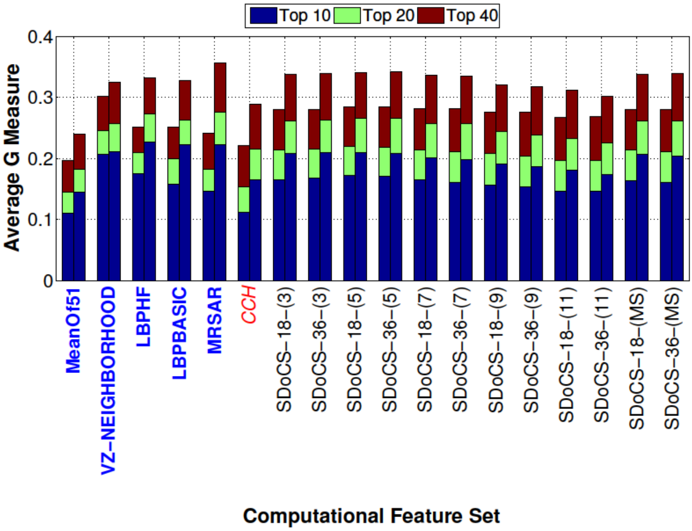

Figure 5: Average G measures of computational features obtained against human ranking data. Each bar shows three different color-coded results for three values of $N \in \{10, 20, 40\}$. In addition, each bar-group shows two resolutions: 1024×1024 (left), and multi-resolution (right). The first five columns (labelled in blue) show the mean and best results obtained using the 51 feature sets tested in [2] at different conditions. The next column shows results obtained using the Chain Code Histogram (CCH). The remaining results labelled in black “SDoCS-A-SL” are results for our new feature set where segment angle θ is quantized into A bins and the segment length $SL \in \{3,5,7,9,11,MS\}$.

5.2 Retrieval-Based Experiment

In this experiment, the 4 best conventional feature sets investigated in [2], namely, VZ-NEIGHBORHOOD [19], MRSAR [21], LBPBASIC [22], LBPHF [23], are utilized as baselines. The average *G* measures obtained using the feature sets are shown in Figure 5 for retrieval sizes of $N \in \{10, 20, 40\}$. It can be seen that: (1) the multi-resolution scheme improves the performance of all these feature sets; and (2) our feature set outperforms all other feature sets except the VZ-NEIGHBORHOOD at the 1024×1024 resolution and outperforms the multi-resolution implementations with the exception of the multi-resolution MRSAR.

6 Conclusions and Future Work

This paper has investigated the importance of three different types of information for the human perception of texture. Two categories were inspired by the types of data commonly used by existing feature measures, namely, power spectra and the short-range HOS (higher order statistics) available from image patches. The third, contour data, was inspired by the fact that the human visual system is extremely adept at extracting these visual cues and that they encode long-range HOS. We conducted an experiment with human observers that showed that for the *Pertex* database contours are the most useful type of data for human texture discrimination.

This result, together with the fact that none of the 51 feature sets examined by Dong et *al*. [1, 2] use HOS beyond 19×19 pixel neighbourhoods, inspired us to develop a new type of texture feature based on representing contours as sets of segments. We refer to this feature set using the snappy title: Spatial Distribution of Contour Segments or SDoCS for short. It is notable as it exploits both the long-range and short-range HOS available from the segment distributions.

We assessed the SDoCS feature set using two tasks. In the pairs-of-pairs task the classifier simply has to judge which of the two pairs differ most. The second task was a retrieval task. However, because a human-derived perceptual similarity matrix [17] was available we were able to fully rank the results. This allowed us to better assess the ability of features to estimate texture similarity but required us to use the "*G*" performance metric [20] that takes into account these rankings.

The results showed that SDoCS outperformed the other feature sets in the pair-of-pairs task and outperformed all but the VZ-NEIGHBORHOOD feature set at the 1024×1024 resolution and the multi-pyramid MRSAR feature set in the ranking task.

We feel that the key point, however, is that we have showed the usefulness of long-range HOS in computing texture similarity and hope that this will inspire other developments of texture features based on such information.

Acknowledgements

We would like to acknowledge the support of the Life Sciences Interface theme of Heriot-Watt University.

References

[1] X. Dong and M. J. Chantler. The Importance of Long-Range Interactions to Texture Similarity. *Proceedings of the 15th International Conference on Computer Analysis of Images and Patterns*, 8047: 425-432, 2013.

[2] X. Dong, T. Methven, and M. J. Chantler. How Well Do Computational Features Perceptually Rank Textures? A Comparative Evaluation. *Proceedings of the ACM 2014 International Conference on Multimedia Retrieval*, 281-288, 2014.

[3] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A*, 7: 923-932, 1990.

[4] MatlabPyrTools-v1.4, http://www.cns.nyu.edu/~lcv/software.php

[5] D. J. Field, A. Hayes and R. F. Hess. Contour integration by the human visual system: evidence for a local "association field". *Vision Research*. 33: 173-193, 1993.

[6] L. Spillmann and J. S. Werner. Long-range interactions in visual perception. *Trends in Neurosciences*. 19: 428-434, 1996.

[7] J. De Winter and J. Wagemans. The awakening of Attneave's sleeping cat: Identification of everyday objects on the basis of straight-line versions of outlines. *Perception*, 37, 2008.

[8] A. V. Oppenheim and J. S. Lim. The Importance of Phase in Signals. *Proceedings of the IEEE*, 69 (5):529-541, 1991.

[9] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6): 679-698, 1986.

[10] A. D. F. Clarke, F. Halley, A. J. Newell, L. D. Griffin and M. J. Chantler. Perceptual Similarity: A Texture Challenge. *Proceedings of British Machine Vision Conference*, 120.1-120.10, 2011.

[11] X. Qin and Y. Yang. Basic Grey level aura matrices: theory and its application to texture synthesis. *Proceedings of the Tenth IEEE International Conference on Computer Vision*, 1: 128-135, 2005.

[12] D. Zhang and G .Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37: 1-19, 2004.

[13] R. C. Gonzalez and R. E. Woods. *Digital Image processing*, Prentice Hall Upper Saddle River. NJ, 2002.

[14] M. B. Dillencourt, H. Samet and M. Tamminen. A general approach to connected-component labeling for arbitrary image representations. *Journal of the ACM*, 39(2): 253-280, 1992.

[15] S. C. Dakin. Orientation variance as a quantifier of structure in texture. *Spatial Vision*, 12: 1-30, 1999.

[16] J. Iivarinen and A. Visa. Shape recognition of irregular objects. *Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling, Proc. SPIE 2904*, 25-32, 1996.

[17] A.D.F. Clarke, X. Dong and M. J. Chantler. Does Free-sorting Provide a Good Estimate of Visual Similarity. *Predicting Perceptions*, 17-20, 2012.

[18] J. B. Tenenbaum, V. de, Silva and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500): 2319-2323, 2000.

[19] M. Varma and A. Zisserman. A Statistical Approach to Material Classification Using Image Patch Exemplars.*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31: 2032-2047, 2009.

[20] R. Fagin, R. Kumar and D. Sivakumar. Comparing Top K Lists. *Proceedings of 14th ACM-SIAM Symposium on Discrete Algorithms*, 28-36, 2003.

[21] J. Mao and A. K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25(2):173-188, 1992.

[22] T. Ojala, M. Pietikäinen and D. Harwood. A Comparative Study of Texture Measures with Classification Based on Feature Distributions. *Pattern Recognition*, 29: 51-59, 1996.

[23] T. Ahonen, J. Matas, C. He and M. Pietikainen. Rotation Invariant Image Description with Local Binary Pattern Histogram Fourier Features. *Scandinavian Conference on Image Analysis*, 61-70, 2009.