# SHORT LOCAL DESCRIPTORS FROM 2D CONNECTED PATTERN SPECTRA

*Petra Bosilj*[1], *Ewa Kijak*[2], *Michael H. F. Wilkinson*[3], *Sébastien Lefèvre*[1]

[1]Université de Bretagne-Sud – IRISA, Vannes, France
[2]Université de Rennes 1 – IRISA, Rennes, France
[3]Johann Bernoulli Institute, University of Groningen, Groningen, The Netherlands

## ABSTRACT

We propose a local region descriptor based on connected pattern spectra, and combined with normalized central moments. The descriptors are calculated for MSER regions of the image, and their performance compared against SIFT. The MSER regions were chosen because they can be efficiently selected by constructing a max-tree, a structure used to calculate both descriptors and region moments. Experiments on the UCID database show an improvement over SIFT in two out of five experimental setups, and comparable performance in two other experiments. The new descriptors are only half the size of SIFT, resulting in 4 times faster query times when performing exact search on descriptor index built from 262 images.

***Index Terms*—** local region descriptors, pattern spectra, max-tree, CBIR

## 1. INTRODUCTION

The goal of content based image retrieval (CBIR) is retrieving images describing the same object or scene as the query from the database. Standard systems consist of keypoint detection, descriptor calculation and storage in an index. Different indexing schemes are used for database search [1–3]. The detection step either selects interest points or interest regions.We focus here on the description part of the system which benefits from powerful local descriptors [4], and use the well-established SIFT descriptors [5] to obtain a baseline CBIR performance on a database. Future work will include comparisons with SIFT extensions which improve performance [6–8].

The proposed local descriptors are based on pattern spectra, commonly used in image analysis and classification [9] and previously used in CBIR as global descriptors [10, 11]. They can be efficiently computed using a mathematical morphology technique known as granulometry [12] on a max-tree hierarchy [13]. This makes them well suited for description of MSER regions [14] which can be detected using the same structure. Extending [10], we compute 2D size-shape pattern spectra locally, and combine them with normalized central moments for the regions of interest. By construction, the produced descriptors are rotation (and translation) invariant and achieve competitive precision with only half the size of SIFT.

We begin by explaining the detection and description parts of the CBIR system, with the focus on how the max-tree is used for both tasks, in Sec. 2. The experimental setup and the database used for performance evaluation are detailed in Sec. 3 with the results analysis offered in Sec. 4. Possible directions for future work are provided in Sec. 5.

The collaboration between the authors was supported by mobility grants from the Université européenne de Bretagne (UEB), French GdR ISIS from CNRS, and an excellence grant EOLE from the Franco-Dutch Network.

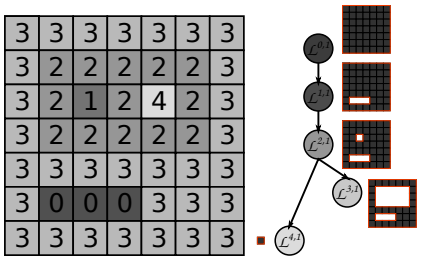


**Fig. 1**: The max-tree for (a) is shown on (b). Nodes are labeled with upper level sets they correspond to, and the regions of the upper level sets are displayed besides the nodes.

## 2. KEYPOINT DETECTION AND DESCRIPTION

### 2.1. Max-tree

Max-tree and min-tree hierarchies [13, 15] of images were used for keypoint detection as well as the feature description. The *upper level set* at level $k$ of an image $I$ is a set of image pixels $p$ with gray level values $f(p)$ higher than a threshold $k$, $\mathcal{L}^k = \{p \in I | f(p) \geq k\}$, where each level set can comprise several connected components.

All the connected components, or the *peak components* $\mathcal{L}^{k,i}$ ($i$ from some index set) of the upper level set $\mathcal{L}^k$ are nested. They form a hierarchy represented by a max-tree (cf. Fig. 1), in which a node $n_{k,i}$ corresponds to the peak component $\mathcal{L}^{k,i}$ at level $k$.

The min-tree can be obtained by considering the *lower level sets* $\{\mathcal{L}_k\}$ of the image or by constructing a max-tree of the inverted image $-I$. These trees are constructed first, after which they are used both for selecting the regions of interest among all the tree regions, and then for calculating the descriptors for the selected regions.

### 2.2. MSER detection

The *Maximally Stable Extremal Regions* (MSER) detector was first introduced by Matas et al. [14]. It responds to bright and dark "blobs" in the image, and is as such complementary to other commonly used detectors [5, 7].

Extremal regions (maximal and minimal) correspond to the peak components of upper and lower level sets $\{\mathcal{L}^{k,i}\}$ and $\{\mathcal{L}_{k,i}\}$, which allows for their detection concurrently with the construction of the max-tree and the min-tree [16]. As such, they are nested, and the local minima of the stability function $q(\mathcal{L}^{k,i})$, calculated for elements of a nested sequence (i.e. nodes on a single path in the tree), correspond to maximally stable regions. This function indicates the rate of growth of the region $\mathcal{L}^{k,i}$ with the decrease of the threshold level $k$. A simplification used by many computer vision libraries

(e.g. VLFeat [17]) for lowering the computation time is used:

$$q(\mathcal{L}^{k,i}) = \frac{|\mathcal{L}^{k-\Delta,i} \setminus \mathcal{L}^{k,i}|}{|\mathcal{L}^{k,i}|}, \quad (1)$$

where $|\cdot|$ denotes cardinality, with $\Delta$ being a detector parameter.

### 2.3. Attributes and filtering

To every node (region) in the tree, we can assign *attributes* pertaining to the characteristics of that node. An attribute $K(\cdot)$ is *increasing* if, for two nested regions $\mathcal{L}^{k,i} \subseteq \mathcal{L}^{l,j}$, its value is always greater for the larger region: $K(\mathcal{L}^{l,j}) \geq K(\mathcal{L}^{k,i})$. Consequently, the attribute value of a node, $K(n_{k,i}) = K(\mathcal{L}^{k,i})$, will be smaller than any of the values of its ancestors. If this property does not hold, the attribute is *nonincreasing*. Out of all nonincreasing attributes, we are here interested in the *strict shape* attributes, which respond only to region shape and are thus invariant to scaling, rotation, and translation [12].

We use here only the attributes that can be computed incrementally, that is the attribute values of the parent nodes can be calculated based on the attribute values of their children, with only examining the new pixels in a region. We use the following attributes:

- *Area:* $A(\mathcal{L}^{k,i})$, the size of the region in pixels, which is an increasing attribute.
- *Binary region moments:* based on raw region moments, we can derive center of mass, covariances, skewness or kurtosis [18]. We will use normalized central moments $n_{1,1}, n_{2,0}, n_{0,2}, n_{0,4}$ and $n_{4,0}$.
- *Corrected noncompactness:* $2\pi(\frac{I(\mathcal{L}^{k,i})}{A(\mathcal{L}^{k,i})^2} + \frac{6}{A(\mathcal{L}^{k,i})})$, an elongation measure used as the (nonincreasing) shape attribute, where $I(\mathcal{L}^{k,i})$ is the moment of inertia of the region. Without the correction factor, $\frac{I(\mathcal{L}^{k,i})}{A(\mathcal{L}^{k,i})^2}$ is equal to the first moment invariant $I = \mu_{2,0} + \mu_{0,2}$ of Hu [19].

Processing a tree, where we decide either to preserve or reject a node by comparing its attribute value to a threshold $K(n_{k,i}) > t$ (or using a more complex criterion), is called a *filtering*. The reader is referred to [10, 12, 13] for more details on the filtering strategies, and attribute filtering based on increasing and nonincreasing attributes.

### 2.4. Granulometries and pattern spectra

When using an increasing attribute, the resulting attribute filtering is an attribute opening (i.e. anti-extensive, increasing and idempotent). A set of such openings for increasing values of the threshold $t$ is called a *size granulometry* and satisfies the absorption property: after an attribute opening, another opening with a lower threshold will have no effect. We can consider a size granulometry as a set of sieves of increasing grades, each passing only details of certain sizes [10].

If we note the amount of detail removed between pairs of consecutive openings, we obtain a *size pattern spectrum*. Introduced by Maragos [9], size pattern spectra are 1D histograms containing the number of pixels or the sum of gray levels for a range of size classes. Rather than repeatedly filtering an image and computing the difference, a connected pattern spectrum can be calculated in a single pass over a max-tree [10, 12]. More importantly for our purposes, it is also possible to compute a histogram over different shape classes (i. e. ranges of shape attribute values), called a *shape-spectrum* [10].

Combining shape and size pattern spectra, we obtain shape-size pattern spectra [10] corresponding to 2D histograms where every bin contains the information about the amount of image detail for a certain size-shape class. A 2-D size-shape *global pattern spectrum* (GPS) is calculated for the whole image. Calculating it for a node will produce a *local pattern spectrum* (LPS), containing only information derived from the region represented by that node.

**Table 1**: Subsets of the UCID database used in experiments.

| | **# categories / examples** | **categories selected** |
|---|---|---|
| *ucid5* | 31 / 5 | all UCID categories with $\geq 5$ examples |
| *ucid4* | 44 / 4 | all UCID categories $\geq 4$ |
| *ucid3* | 77 / 3 | all UCID categories $\geq 3$ |
| *ucid2* | 137 / 2 | all UCID categories $\geq 2$ |
| *ucid1* | 262 / 1 | all UCID categories |

Previous work [10,11] and our own experiments suggest that the lower attribute values carry more information; thus, a logarithmic binning is used for both attributes. However, determining the bin $c$ corresponding to the value $v$ of an attribute is not trivial, and depends on several parameters:

$$c = \log_{\sqrt[N_{bins}]{SV}}(v\frac{SV}{UB}), \quad (2)$$

where $N_{bins}$ is the number of bins used, and $SV$ the scale value at which the LPS is computed. Attribute values higher than the upper bound $UB$ are discarded (both attributes used have a minimal value of 1). An in-depth discussion of Eq. (2) and the experiments supporting parameter choice is offered in [20],while we only give the final parameter choices here.

We set $N_{bins} = 10$ for the area attribute, and $N_{bins} = 6$ for noncompactness, yielding a 60-bin spectrum. We also found that using an even smaller spectrum does not necessarily decrease performance (cf. Sec. 3 for the comparison with a $9 \times 6$ spectrum), so a smaller spectrum can be used if a shorter descriptor is required.

For the noncompactness, $UB = 53$ (or 57 for the $9 \times 6$ version) is also used for $SV$. These values are similar to the ones used for GPS in CBIR (52 in [10] and 53 in [11]). The area of each MSER is used as $UB$ for the area attribute. In case of GPS, the image area can be used for $SV$ [10, 11], but an optimal $SV$ which would work well for all of the selected regions is an additional parameter. As tuning the other parameters is already not trivial (cf. [20] for a discussion), we chose to present the preliminary results here using a descriptor with relative $SV$ equal to the region area. This construction preserves only the rotation and translation invariance properties of [10, 11], while a scale invariant version is examined in [20].

### 2.5. Algorithm

The system was implemented in C++. The max-tree structure was used for both MSER detection and keypoint description. The non-recursive max-tree algorithm of [16] was used. This allows concurrent computation of the MSER stability function (Eq. (1)), the area attribute and moment of inertia, and the MSER. In order to distinguish between descriptors based on dark and light pattern spectra, an indicator value 2 is appended to every maximal MSER descriptor, and 0 for the minimal MSERs. The complete method is as follows:

- Compute the max-tree and min-tree according to [16].
- As the tree is built, compute the area $A(\cdot)$, moment of inertia $I(\cdot)$ and the stability function $q(\cdot)$ for each node $n_{k,i}$.
- During the tree computation, select the local minima of $q(\mathcal{L}^{k,i})$ and $q(\mathcal{L}_{k,i})$, forming the sets of maximal and minimal MSER regions $\{maxMser\}$ and $\{minMser\}$.

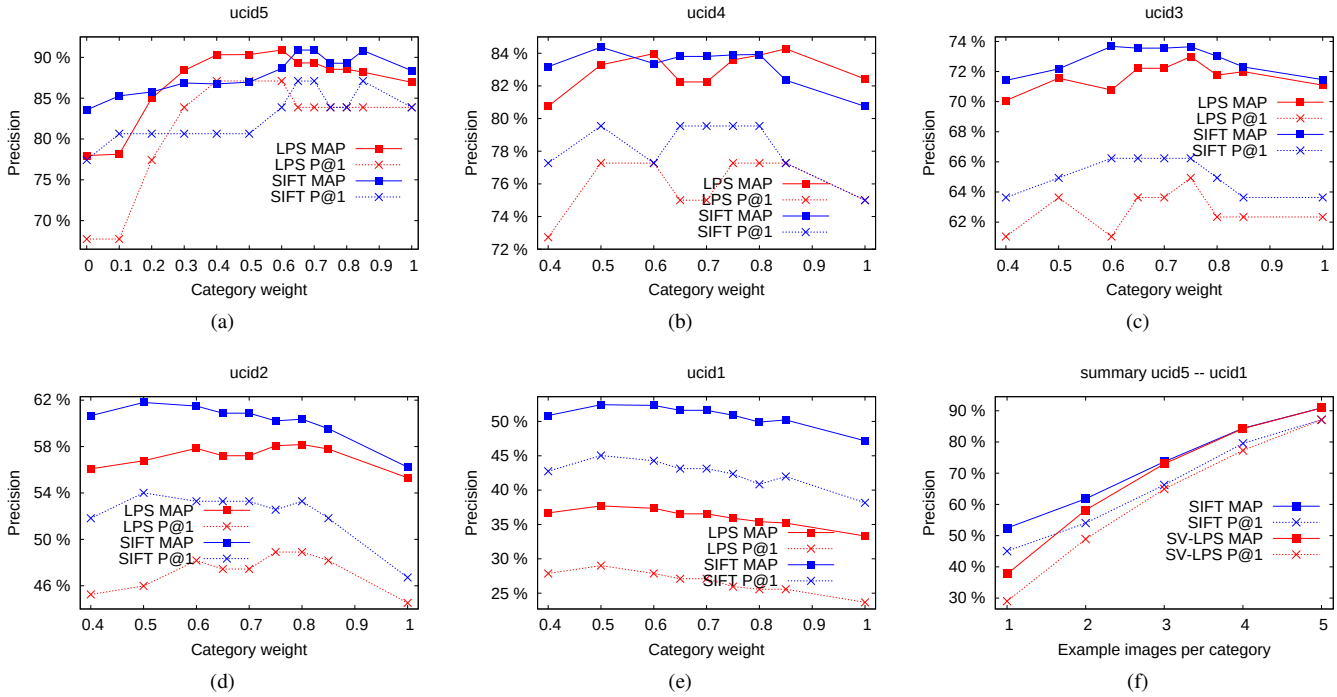


**Fig. 2**: The mean average precision (MAP) and precision at 1 (P@1) for *ucid5–ucid1* datasets for varying category weights are shown in (a)–(e). These results are summarized in (f), with only the results for the optimal weight displayed for every dataset.

- For each node $n_{k,i} \in \{maxMser\}$ and the corresponding region $\mathcal{L}^{k,i}$, examine all the nodes $m_{l,j}$ in the subtree:
  - Set the pattern spectra matrix $S_{n_{k,i}}$ to zero
    - Compute size class $r$ from the area.
    - Compute shape class $s$ from the corrected non-compactness.
    - Compute the gray level difference $\delta_h$ between the node $m_{l,j}$ and its parent (contrast with background), and add $\delta_h A(\mathcal{L}^{k,i})$ to $S_{n_{k,i}}(r,s)$.
  - Interpret the matrix $S_{n_{k,i}}$ as a descriptor vector $D(n_{k,i})$, and append the values of $A(\mathcal{L}^{k,i})$, $n_{1,1}(\mathcal{L}^{k,i})$, $n_{2,0}(\mathcal{L}^{k,i})$, $n_{0,2}(\mathcal{L}^{k,i})$, $n_{0,4}(\mathcal{L}^{k,i})$ and $n_{4,0}(\mathcal{L}^{k,i})$ to $D(n_{k,i})$.
  - Append an indicator value 2 to the descriptor $D(n_{k,i})$.
- Do the same for all the nodes $n_{k,j} \in \{minMser\}$ (appending indicator value 0).
- In addition to all the MSER descriptors, add both global pattern spectra [11] corresponding to the whole image in the collection of descriptors for the image.

The resulting descriptors will have the length of 66, as we are combining a pattern spectrum of length 60, an indicator value depending on if the feature came from the max-tree or the min-tree, and 5 different normalized central moments.

## 3. DATABASE AND EXPERIMENTAL SETUP

Since the large collections of high dimensional data suffer from the "curse of dimensionality", it is needed to use approximate search and indexing schemes such as [1–3] in large scale CBIR. Here, we want to evaluate the performance of the new descriptor without the side-effects of approximate search, so we designed an experimental setup performing exact search and evaluating descriptor performance. The performance of LPS descriptors is compared to SIFT [5].

Different subsets of the *UCID* database [21] were used in the experiments. The whole database contains 1338 images of size $512 \times 384$ pixels in 262 categories of different sizes. To equalize the database entry sizes as much as possible, the number of examples per category is constant for each database subset. The chosen subsets allow observing the effects of increased database size and decreasing number of example images (subsets used detailed in Tab. 1). Only the required number of database images was taken from larger categories in order provided by the ground truth.

For all the database images, the MSER keypoint detection is performed followed by descriptor calculation (LPS or SIFT). The descriptors from all the images of the same category make the database entry for that category, with no difference made between descriptors coming from different images. A KD-Tree index [22] is built based on the category descriptors, and saved for performing the queries using the FLANN library [23].

We then perform a query with a single image for every database category. Keypoints are detected and their descriptors calculated for the query image. The index performs a kNN search ($k = 7$) with each descriptor from the query image. All of the neighbors will cast a vote for the category they belong to as:

$$vote(cat(d_i)) = \frac{1}{(L_1(d_i, q_j) + 0.1) \times |cat(d_i)|^w}. \qquad (3)$$

Here, $q_j$ is the $j$-th query descriptor and $d_i$ the $i$-th nearest neighbor for that descriptor. $L_1(d_i, q_j)$ refers to the distance between these two descriptors and $cat(d_i)$ to the database category of $d_i$, with $|cat(d_i)|$ being the number of descriptors. Finally, $w$ determines the weight with which the category size will contribute.

The categories are sorted according to their vote score and examined in order to evaluate descriptor performance. The measures we

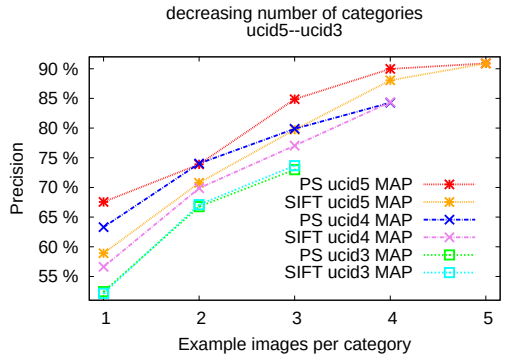


**Fig. 3**: Summarized experimental results on *ucid5* (using 5–1 examples per category), *ucid4* (4–1 examples) and *ucid3* (3–1 examples). Only the highest precision per dataset is shown.

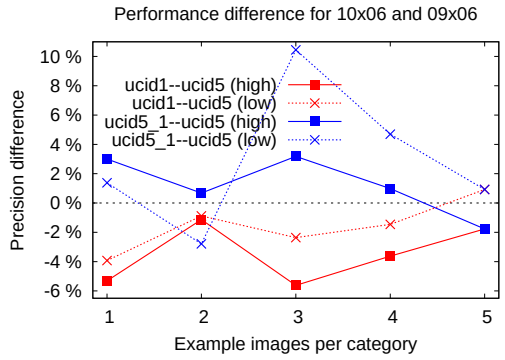


**Fig. 4**: A comparison between LPS using $10 \times 6$ and $9 \times 6$ binning. The difference in highest and lowest achieved precision (for $w \in [0.0, 1.0]$) between the two descriptor versions is shown for the examined datasets (positive difference in favor of $10 \times 6$, and negative for $9 \times 6$).

used are mean average precision (MAP) and precision at one (P@1). All experiments were performed for a $w$ from range $[0.0, 1.0]$ (like in Fig. 2(a)), but to carry out an unbiased comparison, only the highest MAP and P@1 are taken into account for each descriptor when the results are summarized.

## 4. ANALYSIS OF THE RESULTS

We compared the performance of SIFT with that of our LPS descriptor. The results for *ucid5* through *ucid1* for a (reduced) range of weights $w$ and the best LPS and MSER parameters are shown in Fig. 2, with a summary in Fig. 2(f). Note that the category weight $w$ disappears as a parameter when the descriptors are aggregated.

Both the number of categories and the number of examples per category influence the performance. To investigate the influence of decreasing only the number of examples, we repeated the experiments on subsets of *ucid5–ucid3*, using less examples per category. These results are shown in Fig. 3. As expected, the performance of both descriptors declines both for increasing the number of categories, and decreasing the number of examples while keeping the category number constant. However, the rate of precision decline w. r. t. number of examples per category looks moderately lower for the LPS than for SIFT, indicating that using less examples with LPS than with SIFT may be sufficient to achieve desired performance.

Based only on Fig. 2(f), our descriptors give comparable results to the SIFT descriptors for *ucid5 – ucid3*, and perform slightly worse for *ucid2*. However, these results should be considered jointly with the experiments summarized in Fig. 3. When we decrease the number of examples in the *ucid5* and *ucid4* datasets (making the classification problem harder), the LPS descriptors clearly outperform SIFT on these datasets. On the last subset, *ucid1*, our method is significantly outperformed by SIFT descriptors. However, this is the dataset with the largest number of categories but only one example per category. It is known that a certain minimal number of examples (growing with the increase in the number of categories) is required, otherwise the results of classification using such a model can depend on chance. Because of this, the results on this subset are not as reliable as the results on *ucid5–ucid2*, and further testing on larger databases (including varying the number of categories for a constant number of examples) has to be done.

We also looked into an alternate set of parameters, producing a shorter descriptor. The performance comparison of a LPS using a $10 \times 6$ and a $9 \times 6$ binning is shown in Fig. 4. The comparison is shown for varying number of example images, on *ucid5–ucid1* datasets as well as on the *ucid5* dataset with a varying number of examples. It can be seen that the best performance achieved is very close for both descriptors (the full lines). The shorter version actually performs better on *ucid5–ucid1*, but does not reach the performance of the longer version when varying the number of examples of *ucid5*. The large positive values of the dashed line also indicate that the shorter version is not as stable as the longer one. This is likely due to the difficulty of adjusting the parameters, as the experiments in [20] also suggest some dependence between parameter settings. However, especially with a more efficient procedure to choose the parameters, this justifies looking into even shorter versions of the descriptor when the performance speed (caused e. g. by a large number of regions used) is an issue.

Apart from their performance, the proposed LPS descriptors have another advantage. In addition to the description calculation process being slightly faster for the pattern spectra than for the SIFT descriptors, our descriptors length is only $52\%$ of the length of SIFT (or less if using a shorter descriptor version). Using the LPS descriptors gives roughly a $4$ times gain in query speed over the SIFT descriptors on an index of the size $262$ (*ucid1* dataset). This suggests that (especially in large scale CBIR systems), we can use more example images in order to enhance the precision, while still performing faster than SIFT.

## 5. CONCLUSIONS AND FUTURE WORK

LPS outperforms SIFT on two datasets while keeping comparable results on the others, all with a descriptor (at most) half the size of SIFT. It is probable that better results could be achieved by combining the current pattern spectrum with pattern spectra based on other shape attributes, like in [11].

This paper presents only a rotation and translation invariant version of the LPS, but further research will focus on the scale invariant version as well as further reduction of descriptor size. Experiments testing for robustness against scale change will be used to compare the performance of the scale-variant and the scale-invariant version. Algorithmic improvements are also being considered [20].

Due to the promising results on the subsets of the UCID dataset, we want to perform more extensive testing, with a large scale CBIR system using approximate search. Comparing LPS using a different distance, or even a divergence (e.g. [24]), should be considered as the $L_1$ distance was designed to compare vectors of scalar values.

## 6. REFERENCES


[1] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive Hashing Scheme Based on P-stable Distributions," in *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, 2004, SCG '04, pp. 253–262.

[2] J. Sivic and A. Zisserman, "Video Google: Efficient visual search of videos," in *Toward Category-Level Object Recognition*, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, Eds., vol. 4170 of *LNCS*, pp. 127–144. Springer, 2006.

[3] H. Lejsek, B. Þ. Jónsson, and L. Amsaleg, "NV-Tree: Nearest Neighbors at the Billion Scale," in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, 2011, ICMR '11, pp. 54:1–54:8.

[4] C. Schmid and R. Mohr, "Object recognition using local characterization and semi-local constraints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 530–534, 1997.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. IEEE, 2004, vol. 2, pp. II–506.

[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[8] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2911–2918.

[9] P. Maragos, "Pattern spectrum and multiscale shape representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 7, pp. 701–716, 1989.

[10] E. R. Urbach, J. B. T. M. Roerdink, and M. H. F. Wilkinson, "Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 2, pp. 272–285, 2007.

[11] F. Tushabe and M. H. F. Wilkinson, "Content-based image retrieval using combined 2D attribute pattern spectra," in *Advances in Multilingual and Multimodal Information Retrieval*, pp. 554–561. Springer, 2008.

[12] E. J. Breen and R. Jones, "Attribute openings, thinnings, and granulometries," *Computer Vision and Image Understanding*, vol. 64, no. 3, pp. 377–389, 1996.

[13] P. Salembier, A. Oliveras, and L. Garrido, "Antiextensive connected operators for image and sequence processing," *Image Processing, IEEE Transactions on*, vol. 7, no. 4, pp. 555–570, 1998.

[14] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[15] R. Jones, "Component trees for image filtering and segmentation," in *IEEE Workshop on Nonlinear Signal and Image Processing, E. Coyle, Ed., Mackinac Island*, 1997.

[16] D. Nistér and H. Stewénius, "Linear time maximally stable extremal regions," in *Computer Vision–ECCV 2008*, pp. 183–196. Springer, 2008.

[17] A. Vedaldi and B Fulkerson, "VLFeat: An Open and Portable Library of Computer Vision Algorithms," `http://www.vlfeat.org/`, 2008.

[18] M. H. F. Wilkinson, "Generalized pattern spectra sensitive to spatial information," in *Pattern Recognition, International Conference on*. IEEE Computer Society, 2002, vol. 1, pp. 10021–10021.

[19] M.-K. Hu, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on*, vol. 8, no. 2, pp. 179–187, 1962.

[20] P. Bosilj, M. H. F. Wilkinson, E. Kijak, and S. Lefèvre, "Local connected 2D pattern spectra descriptors applied to CBIR systems," To appear in Proc. Int. Symp. Math. Morphology (ISMM) 2015.

[21] G. Schaefer and M. Stich, "UCID: An Uncompressed Colour Image Database," in *Electronic Imaging 2004*. International Society for Optics and Photonics, 2003, pp. 472–480.

[22] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 3, pp. 209–226, 1977.

[23] M. Muja and D. G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," in *International Conference on Computer Vision Theory and Application VISSAPP'09)*. 2009, pp. 331–340, INSTICC Press.

[24] E. Mwebaze, P. Schneider, F.-M. Schleif, J. R. Aduwo, J. A. Quinn, S. Haase, T. Villmann, and M. Biehl, "Divergence-based classification in learning vector quantization," *Neurocomputing*, vol. 74, no. 9, pp. 1429–1435, 2011.