

Research Article

Open Access

Petra Bosilj*, Michael H. F. Wilkinson, Ewa Kijak, and Sébastien Lefèvre

Local 2D Pattern Spectra as Connected Region Descriptors

DOI 10.1515/mathm-2016-0011

Received July 20, 2015; accepted February 11, 2016

Abstract: We validate the usage of augmented 2D shape-size pattern spectra, calculated on arbitrary connected regions. The evaluation is performed on MSER regions and competitive performance with SIFT descriptors achieved in a simple retrieval system, by combining the local pattern spectra with normalized central moments. An additional advantage of the proposed descriptors is their size: being half the size of SIFT, they can handle larger databases in a time-efficient manner. We focus in this paper on presenting the challenges faced when transitioning from global pattern spectra to the local ones. An exhaustive study on the parameters and the properties of the newly constructed descriptor is offered, as well as performance results from preliminary experiments, validating the usage of the descriptor. We also consider possible improvements to the quality and computation efficiency of the proposed local descriptors.

Keywords: shape-size pattern spectra, granulometries, max-tree, region descriptors, CBIR

1 Introduction

Pattern spectra are histogram-like structures originating from mathematical morphology, commonly used for image analysis and classification [15], and contain the information on the distribution of sizes and shapes of image components. They can be efficiently computed using a technique known as granulometry [6] on a max-tree and min-tree hierarchy [11, 23].

We study here the 2D pattern spectra, targeting applications in Content Based Image Retrieval (CBIR) in which the aim is to retrieve the database images describing the same object or scene as the query. Previous success in using the pattern spectra as image descriptors computed at the global [27, 28] or pixel scale (known as DMP [3] or DAP [7, 22]) convinced us to investigate their behavior as local descriptors. Two versions of the descriptor are examined - a scale invariant version and a version that is only rotation invariant. This paper is an extended version of [5], where the primary focus was on the parameters of the method. Additional parameter examination is offered, as well as a more comprehensive validation. The experiments presented in the companion paper [4] are added and extended to include both versions of the descriptor. Additionally, first results regarding the descriptor performance under strong scale changes are also presented.

Standard CBIR systems based on local descriptors consist of region detection, calculation of descriptors and their storage in an index. Different indexing schemes are used to perform large scale database search [8, 13, 26], but all need powerful local descriptors to achieve good performance [25]. To construct such a descriptor, we extend [28] and compute 2D size-shape pattern spectra locally while keeping the good characteristics of the global version (scale, translation and rotation invariance, and computation efficiency). However, to evaluate the quality and properties of our proposed local pattern spectra (LPS) descriptors, we need to re-examine the parameters used with global pattern spectra as well as evaluate the effect of the new parameters introduced by the local descriptor scheme.

*Corresponding Author: **Petra Bosilj, Sébastien Lefèvre:** Université de Bretagne-Sud – IRISA, Vannes, France

Michael H. F. Wilkinson: Johann Bernoulli Institute, University of Groningen, Groningen, The Netherlands

Ewa Kijak: Université de Rennes 1 – IRISA, Rennes, France

We evaluate our descriptors on the MSER regions [16] as they can also be computed on a max-tree [21], using the well-established SIFT descriptors [14] to obtain a baseline CBIR performance. Future work will include comparisons with SIFT extensions which improve performance [1, 2, 12]. A competitive precision is achieved when combining the local pattern spectra with normalized central moments. An additional advantage of the descriptor is that it works directly with regions of arbitrary shapes and can thus include shape information, unavailable to e.g. SIFT descriptor. The produced descriptor is also calculated faster than SIFT for the MSER regions, and is only half the size of SIFT resulting in shorter training and retrieval times.

The goal of this paper is to give an overview of choices and challenges faced when reworking a global pattern spectrum into a local one, as well as present the performance of the proposed descriptor in an experimental setup used to obtain preliminary results and validate the usage of local 2D pattern spectra descriptors. The paper begins with presenting the background notions in Sec. 2, with the focus on how the max-tree is used throughout the CBIR system. The experimental framework used to tune and evaluate the descriptors is explained in Sec. 3. The first contribution, examining the properties of the proposed LPS descriptor through the influence of parameters used, can be found in Sec. 4. Section 5 presents the general descriptor performance as well as the performance on a database featuring strong scale changes. Remarks on possible improvements to the efficiency of LPS computation are given in Sec. 6. Finally, the conclusions are drawn and directions for future work offered in Sec. 7.

2 Background

2.1 Max-tree

The concept of min and max-trees [11, 23] is here central for keypoint detection as well as the calculation of feature descriptors. We recall their definition using the *upper* and *lower level sets* of an image, e.g. sets of image pixels p with gray level values $f(p)$ respectively higher and lower than a threshold k .

Given a level k of an image I , each level set is defined as $\mathcal{L}^k = \{p \in I | f(p) \geq k\}$ for the max-tree, or $\mathcal{L}_k = \{p \in I | f(p) \leq k\}$ for the min-tree. The connected components (also called the *peak components*) $\mathcal{L}^{k,i}$ (i from some index set) of upper level sets are nested and form a hierarchy called a max-tree (an example is shown in Fig. 1). The peak components of the lower level sets, $\mathcal{L}_{k,i}$, are also nested and form a hierarchy called a min-tree. However, the min-tree is usually built as a max-tree of the inverted image $-I$.

2.2 MSER detection

Peak components of the upper and lower level sets $\{\mathcal{L}^{k,i}\}$ and $\{\mathcal{L}_{k,i}\}$ coincide with the maximal and minimal extremal regions in the context of *Maximally Stable Extremal Regions* (MSEr) detector introduced by Matas et al. [16]. The detected regions correspond to bright and dark “blobs” in the image and can be extracted while building the max-tree and the min-tree [21].

Extraction of MSEr relies on the stability function $q(\mathcal{L}^{k,i})$, which measures the rate of growth of the region w.r.t. the change of the threshold level k . It is computed for all the elements of nested sequences, and the local minima of this function correspond to the maximally stable regions.

We use here a simplification commonly adopted by many computer vision libraries (e.g. VLFeat [30]):

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

where the area is denoted by $A(\cdot)$ and Δ is a parameter of the detector. Additional parameters control the allowed region size, limit the appearance of too similar regions and impose a lower limit on the stability score. The implementation parameters were set by comparing the performance to the MSEr implementation provided for [18]. For setting these parameters, we used the *viewpoint* dataset and the same measures (repeatability and matching score) provided in [18].

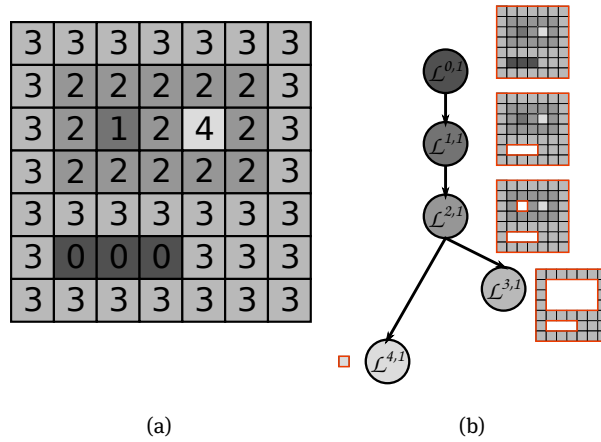


Figure 1: The max-tree for (a) is shown on (b). Nodes are labeled with upper level sets they correspond to, and the regions of the upper level sets are displayed besides the nodes.

2.3 Attributes and Filtering

Region characteristics can be captured by assigning them *attributes* measuring the interesting aspects of the regions. *Increasing* attributes $K(\cdot)$ give increasing values when calculated on a nested sequence of regions, otherwise they are *nonincreasing*. A value of an increasing attribute on a tree region, $K(\mathcal{L}^{k,i})$, will be greater than the value of that attribute for any of the regions descendants.

Increasing attributes are usually a measure of the *size* of the region. We will simply use the *area* (in pixels) of the region, $A(\mathcal{L}^{k,i})$, as the size attribute. *Strict shape* attributes are the nonincreasing attributes dependent only on the region shape, thus invariant to scaling, rotation and translation [6]. To indicate the shape of a region, we use an elongation measure called *corrected noncompactness*:

$$NC(\mathcal{L}^{k,i}) = 2\pi \left(\frac{I(\mathcal{L}^{k,i})}{A(\mathcal{L}^{k,i})^2} + \frac{1}{6A(\mathcal{L}^{k,i})} \right). \quad (2)$$

$I(\mathcal{L}^{k,i})$ is here the moment of inertia of the region, and the term $\frac{I(\mathcal{L}^{k,i})}{A(\mathcal{L}^{k,i})^2}$ without the correction is equal to the first moment invariant of Hu [10] $I = \mu_{2,0} + \mu_{0,2}$. The correction factor appears when transitioning from the original formula in the continuous space to the discrete image space [31].

We also directly use the normalized central moments $n_{1,1}$, $n_{2,0}$, $n_{0,2}$, $n_{0,4}$ and $n_{4,0}$ of the considered regions. These, and more attributes (e.g. center of mass, covariances, skewness or kurtosis [32]), can be derived from raw region moments.

When the tree is further processed by comparing the region attribute values to a threshold t (or using a more complex criterion), and making a decision to preserve or reject a region based on the attribute value, we are performing an *attribute filtering*. Filtering strategies can be divided into *pruning* and *nonpruning*. The pruning strategies will remove all the descendants of a node $\mathcal{L}^{k,i}$ if the node $\mathcal{L}^{k,i}$ is removed. The nonpruning strategies preserve the children of a removed node $n_{k,i}$ and make them the children of the parent of $\mathcal{L}^{k,i}$. Attribute filterings based on increasing and nonincreasing (and especially shape) attributes are extensively discussed by Breen and Jones [6]. For more details on the filtering strategies, the reader is referred to [23] where different filtering rules were first introduced, and [28, 33] which propose and analyze further nonpruning rules.

2.4 Granulometries and Global Pattern Spectra

Attribute opening is a specific kind of attribute filtering, in which the attribute used is increasing and can always be realized by pruning the tree. Such a transformation is anti-extensive, increasing and idempotent. A *size granulometry* can be computed from a series of such openings, using increasing values for the threshold t . This series also satisfies the absorption property, since applying an opening with $t' < t$ will have no effect on an image already filtered with an opening using the threshold t . In other words, a size granulometry can be seen as a set of sieves of increasing grades, each letting only details of certain sizes pass through [28].

Instead of focusing on the details remaining, one can consider the amount of detail removed between consecutive openings. Such an analysis, introduced by Maragos [15] under the name *size pattern spectra*, produces a 1D histogram containing, for each size class or filtering residue, its Lebesgue measure (i. e. the number of pixels in the binary case or the sum of gray levels in the grayscale case). When constructing an attribute filter from nonincreasing shape attributes, we want to filter out all the components not satisfying the criteria while not removing any that do satisfy it. Because of this, a nonpruning strategy has to be used. These operators are anti-extensive and idempotent, making them attribute thinnings on the tree. When the *subtractive* [28] rule is used, preserving the gray level difference between nodes and their parents, a set of such attribute thinnings with an increasing threshold t will also satisfy the absorption property, and form a *shape granulometry* [6, 28]. A *shape pattern spectrum* [28] can then be viewed as a 1D histogram of the distribution of image detail over a range of shape classes. Finally, shape and size pattern spectra can be combined to build *shape-size pattern spectra* [28], 2D histograms where the amount of image detail for the different shape-size classes is stored in dedicated 2D bins.

Previous work [27, 28] as well as our own experiments suggest that the lower attribute values carry more information. Thus, a logarithmic binning is used for both attributes, producing higher resolution bins for low attribute values. Let v be the attribute value for one of the attributes, N_b the total desired *number of bins* and m the *upper bound* for that attribute (which can be the maximal attribute value in the hierarchy, or a smaller value if we decide to ignore attribute values above a certain threshold). If the minimal value for the attribute is 1 (as with both area and the corrected noncompactness), the base for the logarithmic binning b , and the final bin c , are determined as:

$$b = \sqrt[N_b]{m}, \quad (3)$$

$$c = \lfloor \log_b v \rfloor \quad (4)$$

Enumerating the bins starting from 1, the i -th bin has the range $[b^{i-1}, b^i]$.

A connected pattern spectrum is efficiently calculated in a single pass over a max-tree [6, 28]. For every region, we calculate both the size attribute $v_1 = A(\mathcal{L}^{k,i})$ and shape attribute $v_2 = NC(\mathcal{L}^{k,i})$, and add the area of the region weighted by its contrast with the parent region δ_h to the spectrum bin $S(c_1, c_2)$. Before using the spectrum as a descriptor, we equalize the sum across all the bins as $S(c_1, c_2) = \sqrt[5]{S(c_1, c_2)}$. More information and discussion about the algorithm used to compute the descriptors is given in Sec. 6.

3 Database and Experimental Setup

To evaluate the retrieval performance of the descriptors without introducing noise in the results with approximate search approaches [13, 26], we chose a relatively small *UCID* database [24], on which we can perform an exact search. The performance of our proposed descriptors is compared to SIFT [14].

The whole *UCID* database contains 1338 images of size 512×384 pixels, divided into 262 unbalanced categories with one query image assigned to each category. After region detection and description, a single database entry for every category is constructed, comprising the descriptors from all the images of that category (the evaluation can thus be interpreted as a classification problem as well). Therefore, to equalize the database entry sizes as much as possible, different subsets of the *UCID* database were used in the experiments, where the number of examples per category is constant for each database subset (the required number of im-

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

	# categories / examples	categories selected
<i>ucid5</i>	31 / 5	all UCID categories with ≥ 5 examples
<i>ucid4</i>	44 / 4	all UCID categories ≥ 4
<i>ucid3</i>	77 / 3	all UCID categories ≥ 3
<i>ucid2</i>	137 / 2	all UCID categories ≥ 2
<i>ucid1</i>	262 / 1	all UCID categories

Table 2: Rescaling on different databases used.

	query	DB #1	DB #2	DB #3	DB #4	DB #5
<i>ucid5r</i>	$\times 1$	$\times 4$	$\times 2$	$\times 1$	$\times 0.5$	$\times 0.25$
<i>ucid5q4</i>	$\times 4$	$\times 1$	$\times 1$	$\times 1$	$\times 1$	$\times 1$
<i>ucid5q2</i>	$\times 2$	$\times 1$	$\times 1$	$\times 1$	$\times 1$	$\times 1$
<i>ucid5q05</i>	$\times 0.5$	$\times 1$	$\times 1$	$\times 1$	$\times 1$	$\times 1$
<i>ucid5q025</i>	$\times 0.25$	$\times 1$	$\times 1$	$\times 1$	$\times 1$	$\times 1$

ages is taken from larger categories in order provided by the ground truth). Table 1 summarizes the subsets of the database used for experiments presented herein. As both the number of categories and the number of examples per category affect the performance, to investigate the influence of changing only the number of example images, further experiments were performed on the subsets of *ucid5*–*ucid3* for a decreasing number of examples per category.

Furthermore, in order to test the influence of scale change on the performance we use the *ucid5r* database, obtained from *ucid5* by upscaling 2 of the database images, and downscaling another 2, while the query and one of the database images are left at the original scale. Additionally, to examine separately the influence of rescaling by different amounts, *ucid5q4*–*ucid5q025* databases are constructed, where only the query is rescaled. The precise scales for each of the rescaled databases is shown in Tab. 2.

A KD-Tree index [9] is built based on the category descriptors, and stored for querying using the FLANN library [19]. We then perform a query with 1 image for every database category. The index performs a kNN search ($k = 7$) with each descriptor of a region detected on the query image. The final category is given through a voting mechanism where each nearest neighbor d_i of a query descriptor q_j will cast a vote for the category $cat(d_i)$ it belongs to:

$$vote(cat(d_i)) = \frac{1}{(L_1(d_i, q_j) + 0.1) \times |cat(d_i)|^{w_{cat}}} \quad (5)$$

$L_1(d_i, q_j)$ refers to the distance between these two descriptors and $|cat(d_i)|$ is the number of descriptors in the category of the i -th nearest neighbor. Finally, w_{cat} is a parameter of the experimental setup. A $k > 1$ is chosen to take into account several nearest neighbors if their distance is very similar. As the vote contribution decreases with the distance, very far neighbors will have a negligible contribution even if considered.

The measures we used are mean average precision (MAP) and precision at one (P@1). Performance for different values of w_{cat} are shown in Fig. 3(d) and Figs. 4(a)–4(e), but for the summarized results, only the performance for the optimal w_{cat} value for each experiment is shown. This choice is made in order to present a fair comparison, and since not all the descriptors reach their peak performance for the same value of w_{cat} . This is additionally justified as this parameter is not present when using an approximate classification scheme.

Table 3: Parameters and their optimal values for the LPS (best alternative parameter choices also given).

symbol	significance	value SI-LPS	value SV-LPS
m_A	upper bound for area	region size	
m_{NC}	upper bound for noncompactness	53 (54, 56)	53 (57)
N_b^A	number of area bins	10	10 (9)
N_b^{NC}	number of noncompactness bins	6	
RS	reference scale for the size attribute	1000	region size
$w(n_{1,1})$	normalized	20	
$w(n_{2,0}), w(n_{0,2}),$ $w(n_{4,0}), w(n_{0,4})$	moment weights	10	

4 Local Pattern Spectra

Local pattern spectra (LPS) are calculated from the selected MSER regions. As the two trees contain different regions, the descriptor for a maximal MSER will only be based on the max-tree, and similarly for the minimal MSERs.

The LPS are calculated like the global ones, except the calculation is done on the corresponding subtree. When calculating the LPS for the MSER region $\mathcal{L}^{k,i}$ in the tree, we only consider the attribute values of the descendants of the node. However, transitioning to the local version of the descriptor will introduce a new parameter influencing the scale invariance property of the descriptors.

To achieve both the desired properties and competitive performance, the proposed descriptor is explained here through examining the experiments used to establish the best parameters. The summary of these parameters, explained individually henceforth, can be found in Tab. 3. Additionally, we consider combining the LPS with normalized central moments and enhancing the performance by adding the global pattern spectra. The choice of measurement region on which the descriptors are calculated in relation to the actual detected region is also discussed.

4.1 Achieving Scale Invariance

When calculating a global pattern spectrum for an entire image, the whole image size is used to determine the base of the logarithmic binning (especially since the database images are usually of the same or similar size [27, 28]). If we choose to determine the binning base for each region separately in the local description scheme and base it directly on the area of that region, the resulting LPS descriptor is not scale invariant.

Let us consider two versions of the same region at different scales, with the area values belonging to the range $[1, m_1]$ and $[1, m_2]$ respectively. The scale invariance property requires that, for a value $v_1 \in [1, m_1]$, the bin c_1 determined in the original scale is the same as the bin c_2 for the value $v_2 = v_1 \frac{m_2}{m_1}$ scaled to the range $[1, m_2]$. However, this is not the case for $m_1 \neq m_2$, as:

$$c_1 = \log_{\sqrt[m_1]{v_1}} v_1 \neq c_2 = \log_{\sqrt[m_2]{v_2}} v_2. \quad (6)$$

Therefore, to ensure the scale invariance, the areas used to determine the binning and the logarithmic base have to be the same for all the regions. This area becomes a parameter of the size attribute in LPS, called the *reference scale* RS .

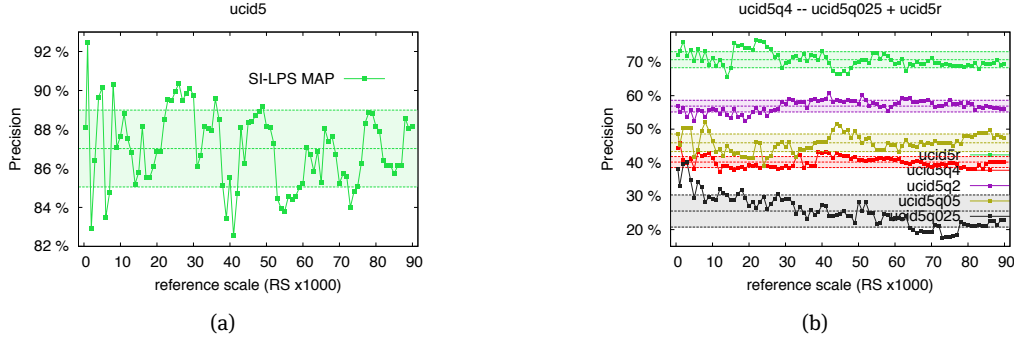


Figure 2: Performance of SI-LPS for a range of reference scales (mean and standard deviation displayed). The performance for the *ucid5* database is shown in (a), while (b) summarizes the influence of this parameter for all the databases with scale changes (listed in Tab. 2).

Using a common scale RS can be seen as rescaling all the regions to the same reference scale, and has two consequences. First, for a region of size $m > M$, the minimal value v of this region that can contribute to the spectrum when using a common binning is such that $v' = v \frac{M}{m} = 1$, and all the (sub)regions with the area smaller than $\frac{m}{M}$ will be ignored. However, some particular regions with a large enough area can still disappear when rescaling. This is the case for long thin objects with the width (along any dimension) small enough to downscale to under 1 pixel. Such regions should be ignored in the pattern spectrum, even if their attribute values fit with the binning. Because of this, we also determine the maximal possible value of the noncompactness attribute for all of the available area bins and use it as a criterion to discard regions.

Second, the minimal area value (1 pixel) of a region of size $m < M$ will be rescaled to the value $v' = \frac{M}{m} > 1$, and the lower area bins at the common scale will be empty. The first area bin c_{\min} that will contain information is then:

$$1 = b^{c_{\min}-1} \frac{m}{M} \rightarrow c_{\min} = \left\lceil \log_b \frac{M}{m} \right\rceil + 1. \quad (7)$$

We compare 2 versions of the descriptor: a) the *scale variant version* (SV-LPS), where the area of each region is used as the reference scale RS , and b) the *scale invariant version* (SI-LPS) where RS is the same for all regions. We test the performance of the SI-LPS for a range of reference scales between 500 and 90000. The results in terms of MAP, as well as their mean and standard deviation are shown in Fig. 2(a). The performance is fairly stable under varying reference scale, with the difference between best and worst performance lower than 10% and a small standard deviation for the chosen range. Some significant local maxima and minima still exist, most likely due to quantisation effects, and should be examined more closely. All further results on *ucid1-ucid5* are obtained using the scale parameter $RS = 1000$ resulting in best performance on *ucid5*. The influence of the reference scale when scale changes are introduced to the database is also analyzed, and shown in Fig. 2(b). This will be discussed together with other experimental results regarding scale invariance under strong scale changes in Sec. 5.2, but clearly demonstrates that the stability under reference scale is not negatively influenced by scale changes in the database.

4.2 Binning Parameters

With the area attribute, the upper bound used, m_A , is simply the size of the region: we can plausibly expect regions of all sizes lower than the size of the region itself to be present in its decomposition.

Examining the values of the noncompactness attribute for several images, we determined that very few regions have high values of this attribute. As such, noncompactness values higher than a certain threshold can be safely ignored. Optimal values m_{NC} for both SV-LPS and SI-LPS were determined by examining the performance of the values close to the ones used in [27, 28]. Similar experiments were done to determine N_b^{NC} and N_b^A . The parameter tuning experiments for the *ucid5* database are shown in Fig. 3.

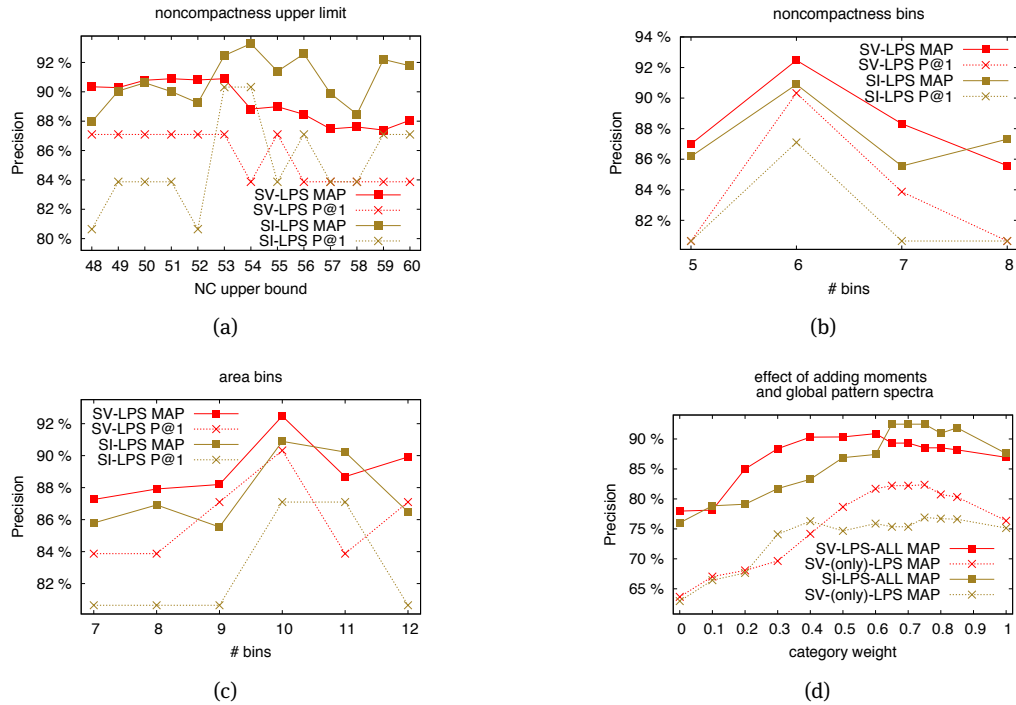


Figure 3: Parameter tuning on *ucid5* database. The effect of varying the upper bound for noncompactness is shown on (a), similar for the amount of noncompactness bins on (b), and the area bins on (c). The effect of adding the moments and indicator value to the descriptor, with the best parameter settings is shown in (d). Note that the global descriptors for the SI-LPS are calculated with the scale value used for the other descriptors, and not using image size.

For both descriptors, we chose $N_b^{NC} = 6$ and $N_b^A = 10$. To choose between several values of m_{NC} performing well on *ucid5*, we compare their performance on *ucid4-ucid1* as well. This was done as the performance for different values of m_{NC} is fairly stable (only about 5% difference for values shown on Fig. 3(a)).

Surprisingly, we also found an alternative set of values for SV-LPS with the lower value of $N_b^A = 9$ but a higher $m_{NC} = 57$. The optimal values as well as the best alternative choices are shown in Tab. 3. As an alternate set of parameters was found producing shorter SV-LPS descriptors, the possibility of further shortening the SI-LPS without the loss in performance should also be investigated.

4.3 Image Moments and Global Pattern Spectra

Five image moments, $n_{1,1}$, $n_{2,0}$, $n_{0,2}$, $n_{0,4}$ and $n_{4,0}$, were appended to a final version of all LPS descriptors (all normalized central moments up to the order 5 were considered). The weights resulting in the best performance (using the L_1 distance) were determined by examining the combination of the LPS and each of the moments separately. This weight is 20 for $n_{1,1}$ and 10 for other moments used. Additionally, an indicator value 2 is added to all the LPS descriptors originating from the max-tree, and 0 for the min-tree, thus additionally increasing the L_1 distance between any minimal and maximal MSERs.

Global pattern spectra on their own achieve MAP around 70% on the *ucid5* dataset. They are added to the list of LPS for every image and treated equally to other local descriptors. Note that they are also calculated on a common *RS* when combined with the SI-LPS. The influence of combining these values with SI-LPS and SV-LPS for the optimal parameter choice is shown in Fig. 3(d).

4.4 Region Size Influence

Before calculating any descriptors in the evaluation framework of Mikolajczyk et al. [17, 18], the region is first approximated by an ellipse with the same corresponding second moments, and then the region size is increased three times to construct a measurement region from the detected region, using affine covariant construction. Only then is the SIFT descriptor calculated on the measurement region using the provided implementation [14].

Since we want to be able to use the max-tree and the min-tree for the pattern spectra calculation, we chose to work with ancestor regions of the detected MSER such that the size of the ancestor is no larger than $\chi A(n_{k,i})$. We determined that, in order to get the same average area increase as in [17, 18], we should use the value $\chi = 7.5$. The reason is that many regions have a much bigger parent region, which is then not considered, and the size increase is often smaller than χ times. This means that the obtained LPS descriptors will also include the shape information (of either the detected region directly or an ancestral region) which gives it an additional advantage when used with detectors returning regions of arbitrary shapes (such as MSER).

5 Results

5.1 Varying the Number of Categories and Examples

We compared the performance of SIFT with that of our LPS descriptors, and both descriptor versions perform closely to SIFT descriptors in the experiments on *ucid1–ucid5* databases. These results, for a (reduced) range of weights w and the best MSER and LPS parameters (as shown in Tab. 3) are shown in Fig. 4, with a summary in Fig. 4(f).

The performance expectantly decreases with the increase of database size and the decrease of the number of examples per category. Further experiments aiming to separately examine the influence of these two factors are shown in Fig. 5, where the experiments on *ucid3–ucid5* were repeated while decreasing the category size. The rate of precision decline w. r. t. the number of examples per category is lower for the both versions of LPS descriptors (cf. Figs. 5(a) and 5(b) and compare to Fig. 5(c)).

When considering the results presented in Fig. 4 and Fig. 5, we can claim that our descriptors outperform the SIFT descriptor on the *ucid4* and *ucid5* databases. Their performance is comparable on the whole database subsets, but further reducing the number of examples clearly shows the advantage of using LPS descriptors on these databases. We can report comparable results with SIFT on the *ucid3* and a slightly worse performance than SIFT on *ucid2* dataset. On the *ucid1* dataset, both our LPS descriptors are significantly outperformed by SIFT. However, it is known that minimal number of examples (growing when more categories are used) is required for classification. As the *ucid1* dataset is the subset with the largest number of categories used, the classification results, using only the example images of this dataset as a model, might depend on chance and are not as reliable as the results on *ucid2–ucid5*.

Besides the performance, it is important to note here that the descriptor is also calculated faster than SIFT for the MSER regions, and that on the largest database subset used, the query speed for LPS is around $4\times$ faster than that for SIFT (when the LPS descriptor of size 66 is used). As a smaller version of SV-LPS was already found, it is likely possible to further shorten the SI-LPS as well and achieve even faster query speeds without a loss in performance.

5.2 Scale Changes

As the UCID database is not very challenging in terms of scale change, further experiments were done after manually rescaling some of the images in the *ucid5* database subset. In one set of experiments, only the query image was rescaled (downscaled or upscaled), corresponding to *ucid5q025–ucid5q4* datasets. Additionally,

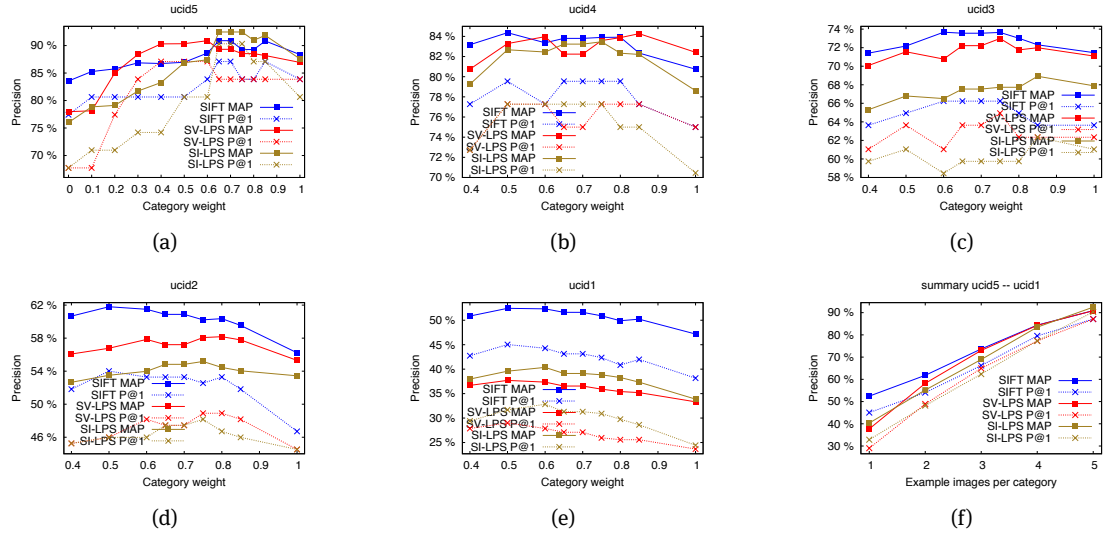


Figure 4: The results for the final version of the descriptors expressed in terms of mean average precision (MAP) and precision at 1 (P@1) for *ucid5–ucid1* dataset for varying category weights are shown in (a)–(e). The results for *ucid5–ucid1* are summarized on (f) (performance shown for optimal weight w_{cat} for every dataset). The complete results for the SV-LPS were previously presented in [4].

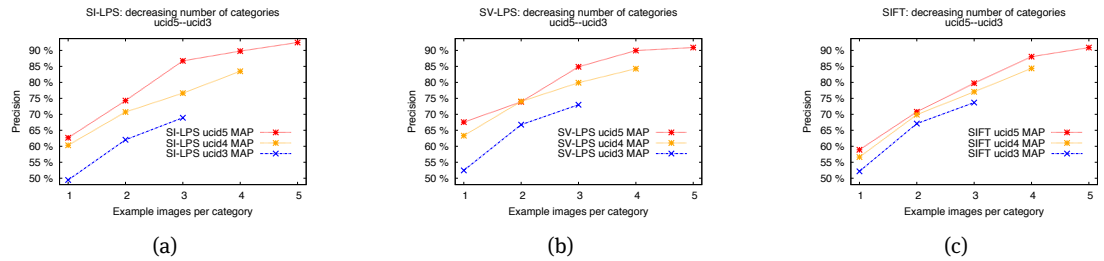


Figure 5: Summarized experimental results on *ucid5* (using 5–1 examples per category), *ucid4* (4–1 examples) and *ucid3* (3–1 examples). Only the highest precision per dataset is shown. The results are shown separately for the three descriptors, with SI-LPS shown in (a), the SV-LPS shown in (b) and SIFT shown in (c) ((b) and (c) results previously shown in [4]).

to examine the influence of introducing different scale changes at once, all the example images were resized by different scale factors in *ucid5r*. All the database subsets with introduced scale changes are listed in Tab. 2.

Before examining the performance on these datasets, we need to validate the choice of the reference scale parameter RS . This is shown for all the rescaled subsets in Fig. 2(b), where it can be seen that the performance on the (composite) *ucid5r* database is in fact more than a combination of the performance contributions when only one type of scale change is introduced. We can conclude, expectedly, that downscaling has a more severe effect on the performance than upscaling as it always results in the loss of image detail. We can also see that a relative stability under the range of reference scales is preserved after introducing scale changes, however the optimal performance is achieved for a reference scale $RS = 22000$. Still, the relative stability under the reference scale change can be seen in Fig. 2(b), and comparing with Fig. 2(a) confirms that using any of the two optima still gives good performance on either of the datasets.

Finally, the performance comparison of LPS and SIFT descriptors for a *ucid5r* database, comprising different scale changes, is shown in Fig. 6. The performance with the best choice of the w parameter of the SI-LPS descriptor comes close to the performance of SIFT in Fig. 6(a). It is also consistently higher than the performance of SV-LPS for all the values of w (and for all the values of RS). The decline in performance on the *ucid5r* database as compared to the database with no rescaling is shown for all descriptors on Fig. 6(b) (for SI-LPS,

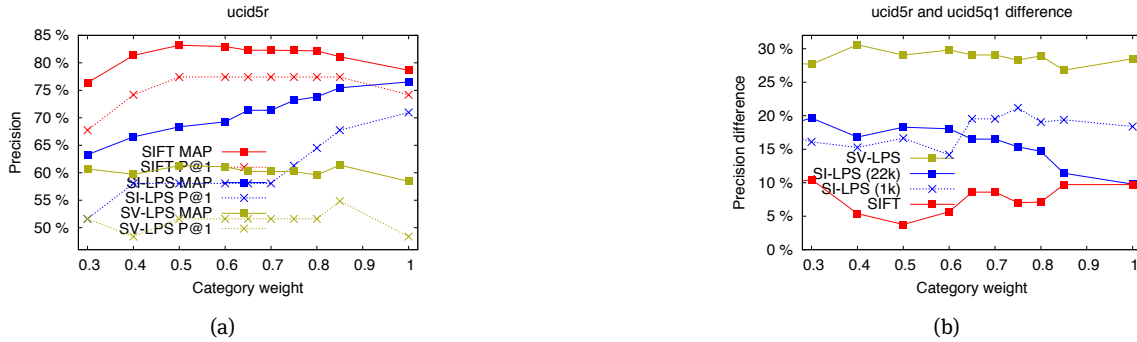


Figure 6: The performance of SI-LPS descriptors using the optimal value of $RS = 22000$ is compared to the performance for SIFT and SV-LPS descriptors on *ucid5r* dataset for a range of w values in (a). The decline in performance when as compared to the performance on the dataset before any rescaling (*ucid5*) for all three descriptors is shown in (b). For the SI-LPS, this difference is shown for both the optimal RS value on *ucid5r*, as well as the optimal value for the *ucid5* dataset, $RS = 1000$.

for both the reference scale best performing on *ucid5r* and *ucid5*). In this figure it is clearly visible that the performance drop is much stronger for the SV-LPS, i.e. that the SI-LPS indeed have scale invariant properties.

6 Remarks on the 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 [21] was used. This allows concurrent computation of the MSER stability function (Eq. (1)), the area attribute and the moment of inertia, and the MSER. The method is as follows:

- Compute the max-tree and min-tree according to [21].
- As the trees are built, compute:
 - attribute values for the nodes of the trees (including region moments, area and shape attribute),
 - local minima of the stability function, forming the sets of MSER regions,
 - global pattern spectra [28].
- For each selected MSER region, repeat the computation of the pattern spectra locally in a sub-tree.
- Combine the region moment values, indicator value 0 or 2 and the pattern spectra to form a LPS descriptor for a MSER region.
- Add both global pattern spectra [27] corresponding to the whole image in the collection of descriptors for the image.

Unlike the calculation of global pattern spectra, the local pattern spectra use the constructed hierarchy but can not be computed concurrently because of different upper limits (for area) and binning scaling value.

However, adopting the scale invariant version to concurrent computation can be considered. While it would sacrifice true scale invariance, if the value RS is used as a reference scale, and we are calculating for a region of size m , we can set the largest bin to be $[b^{\lceil \log_b m \rceil - 1}, b^{\lceil \log_b m \rceil}]$, with the smallest bin having the upper bound $b^{\lceil \log_b m \rceil - N_b}$. While not all the values from the whole range of the largest bin will be possible for all the regions, the bin values of the children can be used directly by their parents. When the upper bound of the largest bin changes, the child values can still be used with discarding the values from the smallest bin: the scale of those details is too low to be considered.

7 Discussion and Conclusion

After successfully applying global pattern spectra in CBIR context [27, 29], we now explore a local region descriptor based on the pattern spectra. On the chosen subsets of the *UCID* database [24], the results obtained were better than when only using global pattern spectra (almost 20% in MAP on *ucid5*), and matched the performance of the SIFT descriptor. The constructed SI-LPS descriptors keep all the invariance properties of the global pattern spectra (translation, rotation and scale invariance).

The proposed 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 half of the length of SIFT. This makes using these descriptors much faster – performing 262 queries on an index of the size 262 (*ucid1* dataset) took 4 times longer using SIFT descriptors. This suggests that (especially in large scale retrieval systems), we can use more example images in order to enhance the precision, while still performing faster than SIFT.

As the performance of the descriptors depends on a lot of parameters, we need to explore a way to determine the optimal parameters automatically. Also, while the LPS descriptors are rotation invariant, enforcing scale invariance introduces an additional parameter. In addition to examining this new parameter closer, both SI-LPS and SV-LPS were evaluated on a database focused on scale changes to determine the value of true scale invariance in such cases, which confirmed additional stability properties of SI-LPS.

It is probable that the results could be even further improved by combining the current LPS with pattern spectra based on other shape attributes, like in [27]. Lastly, the L_1 distance, designed to compare vectors of scalar values, is not the best choice for comparing histogram-like structures. Using different distances, or even divergences (e.g. [20]) which take into account the nature of the descriptor should also improve the performance.

Acknowledgement: 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.

References

- [1] 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*, pages 2911–2918. IEEE, 2012.
- [2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3):346–359, 2008.
- [3] J. A. Benediktsson, M. Pesaresi, and K. Arnason. Classification and Feature Extraction for Remote Sensing Images from Urban Areas based on Morphological Transformations. *IEEE Transactions on Geoscience and Remote Sensing*, 41(9):1940–1949, 2003.
- [4] P. Bosilj, E. Kijak, M. H. F. Wilkinson, and S. Lefèvre. Short local descriptors from 2D connected pattern spectra. To appear in ICIP 2015.
- [5] Petra Bosilj, Michael HF Wilkinson, Ewa Kijak, and Sébastien Lefèvre. Local 2d pattern spectra as connected region descriptors. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 182–193. Springer, 2015.
- [6] E. J. Breen and R. Jones. Attribute openings, thinnings, and granulometries. *Computer Vision and Image Understanding*, 64(3):377–389, 1996.
- [7] K. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone. Morphological Attribute Profiles for the Analysis of Very High Resolution Images. *IEEE Transactions on Geoscience and Remote Sensing*, 48(10):3747–3762, 2010.
- [8] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive Hashing Scheme Based on P-stable Distributions. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry, SCG '04*, pages 253–262, 2004.
- [9] 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)*, 3(3):209–226, 1977.
- [10] M.-K. Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962.

- [11] R. Jones. Component trees for image filtering and segmentation. In *IEEE Workshop on Nonlinear Signal and Image Processing*, E. Coyle, Ed., Mackinac Island, 1997.
- [12] 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*, volume 2, pages II–506. IEEE, 2004.
- [13] H. Lejsek, B. P. Jónsson, and L. Amsaleg. NV-Tree: Nearest Neighbors at the Billion Scale. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR '11, pages 54:1–54:8, 2011.
- [14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [15] P. Maragos. Pattern spectrum and multiscale shape representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7):701–716, 1989.
- [16] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.
- [17] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.
- [18] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International journal of computer vision*, 65(1-2):43–72, 2005.
- [19] 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*, pages 331–340. INSTICC Press, 2009.
- [20] 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*, 74(9):1429–1435, 2011.
- [21] D. Nistér and H. Stewénius. Linear time maximally stable extremal regions. In *Computer Vision—ECCV 2008*, pages 183–196. Springer, 2008.
- [22] G. K. Ouzounis, M. Pesaresi, and P. Soille. Differential Area Profiles: Decomposition Properties and Efficient Computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1533–1548, 2012.
- [23] P. Salembier, A. Oliveras, and L. Garrido. Antiextensive connected operators for image and sequence processing. *Image Processing, IEEE Transactions on*, 7(4):555–570, 1998.
- [24] G. Schaefer and M. Stich. UCID: An Uncompressed Colour Image Database. In *Electronic Imaging 2004*, pages 472–480. International Society for Optics and Photonics, 2003.
- [25] C. Schmid and R. Mohr. Object recognition using local characterization and semi-local constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, 1997.
- [26] J. Sivic and A. Zisserman. Video Google: Efficient visual search of videos. In J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, editors, *Toward Category-Level Object Recognition*, volume 4170 of *LNCS*, pages 127–144. Springer, 2006.
- [27] 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*, pages 554–561. Springer, 2008.
- [28] 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*, 29(2):272–285, 2007.
- [29] E. R. Urbach and M. H. F. Wilkinson. Shape-only granulometries and grey-scale shape filters. In *Proc. Int. Symp. Math. Morphology (ISMM)*, volume 2002, pages 305–314, 2002.
- [30] A. Vedaldi and B. Fulkerson. VLFeat: An Open and Portable Library of Computer Vision Algorithms. <http://www.vlfeat.org/>, 2008.
- [31] M. A. Westenberg, J. B. T. M. Roerdink, and M. H. F. Wilkinson. Volumetric Attribute Filtering and Interactive Visualization using the Max-Tree Representation. *IEEE Trans. Image Proc.*, 16:2943–2952, 2007.
- [32] M. H. F. Wilkinson. Generalized pattern spectra sensitive to spatial information. In *Pattern Recognition, International Conference on*, volume 1, pages 10021–10021. IEEE Computer Society, 2002.
- [33] Y. Xu, T. Géraud, and L. Najman. Morphological filtering in shape spaces: Applications using tree-based image representations. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 485–488. IEEE, 2012.