Short Local Descriptors from 2D Connected Pattern Spectra
Petra Bosilj, Ewa Kijak, Michael H.F. Wilkinson, Sébastien Lefèvre

To cite this version:
Petra Bosilj, Ewa Kijak, Michael H.F. Wilkinson, Sébastien Lefèvre. Short Local Descriptors from 2D Connected Pattern Spectra. IEEE International Conference on Image Processing, 2015, Quebec City, Canada. hal-01134071

HAL Id: hal-01134071
https://inria.hal.science/hal-01134071
Submitted on 21 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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 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. This produces a rotation invariant descriptor half the size of SIFT which achieves competitive precision.

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.

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. KEYPIN 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 $k$, $L^k = \{p \in I | f(p) \geq k\}$, where each level set can comprise several connected components.

All the connected components, or the peak components $L^{k,i}$ ($i$ from some index set) of the upper level set $L^k$ 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 $L^{k,i}$ at level $k$. The min-tree can be obtained by considering the lower level sets $\{L_k\}$ 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 $\{L^{k,i}\}$ and $\{L_{k,i}\}$, 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 $q(L_{k,i})$, 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 $L_{k,i}$ with the decrease of the threshold level $k$. A simplification used by many computer vision libraries

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.
(e.g. VLFeat [17]) for lowering the computation time is used:
\[
q(L^{k,i}) = \frac{|L^{k,\Delta,i} \setminus L^{k,i}|}{|L^{k,i}|},
\]
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 \(L^{k,i} \subseteq L^{k,j}\), its value is always greater for the larger region: \(K(L^{k,j}) \geq K(L^{k,i})\). Consequently, the attribute value of a node, \(K(\{n_k,i\}) = K(L^{k,i})\), 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**: \(A(L^{k,i})\), 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 \(\mu_{n_1,1}, \mu_{n_2,0}, \mu_{0,2}, \mu_{0,4}\) and \(\mu_{4,0}\).
- **Corrected noncompactness**: \(2\pi\sqrt{(\frac{I(L^{k,i})}{A(L^{k,i})} + \frac{A(L^{k,i})}{6}}\), an elongation measure used as the (nonincreasing) shape attribute, where \(I(L^{k,i})\) is the moment of inertia of the region.

Without the correction factor, \(\frac{I(L^{k,i})}{A(L^{k,i})}\) 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 granularity 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

<table>
<thead>
<tr>
<th>Table 1: Subsets of the UCID database used in experiments.</th>
</tr>
</thead>
<tbody>
<tr>
<td># categories / examples</td>
</tr>
<tr>
<td>ucid5</td>
</tr>
<tr>
<td>ucid4</td>
</tr>
<tr>
<td>ucid3</td>
</tr>
<tr>
<td>ucid2</td>
</tr>
<tr>
<td>ucid1</td>
</tr>
</tbody>
</table>

will produce a local pattern spectrum (LPS), containing only information derived from the region represented by that node.

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:

\[
c = \log_{\frac{N_{bins}}{SV}}(\frac{SV}{UB}),
\]

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 a \(9 \times 6\) spectrum causes only a slight decrease in performance (cf. Sec. 3 for the comparison), so a smaller spectrum can be used if an even shorter descriptor is required.

For the noncompactness, \(UB = 53\) (or 56 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]. To keep the scale invariance property of GPS, a same \(SV\) should be used for all LPS. However, it is hard to find an optimal \(SV\), which would work well for all selected regions. By using a relative \(SV\) equal to the region area, we lose scale invariance but achieve better precision in all performed experiments (see [20] for a discussion). The LPS remain both rotation and translation invariant, like in [10,11].

### 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 the 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 \(q(L^{k,i})\), forming the sets of maximal and minimal MSER regions \(\{\text{maxMSER}\}\) and \(\{\text{minMSER}\}\).
For each node $n_{k,i} \in \{\text{maxMser}\}$ and the corresponding region $L^{k,i}$, 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 $\delta_l$ between the node $m_{l,j}$ and its parent, and add $\delta_l A(L^{k,i})$ 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 $A(L^{k,i}), n_{1,1}(L^{k,i}), n_{2,0}(L^{k,i}), n_{0,2}(L^{k,i}), n_{2,4}(L^{k,i})$ and $n_{4,0}(L^{k,i})$ to $D(n_{k,i})$.
- Append an indicator value 2 to the descriptor $D(n_{k,i})$.

Do the same for all the nodes $n_{k,i,j} \in \{\text{minMser}\}$ (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 1 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:

$$vote(\text{cat}(d_i)) = \frac{100}{(L_1(d_i, q_j) + 0.1) \times |\text{cat}(d_i)|^w}.$$  \hspace{1cm} (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 $\text{cat}(d_i)$ to the database category of $d_i$, with $|\text{cat}(d_i)|$ being the number of descriptors. Finally, $w$ determines the weight with which the category size will contribute.

Five categories with the highest vote scores are examined in order to evaluate descriptor performance. The measures we are used
of both descriptors declines both for increasing the number of categories, and decreasing the number of examples while keeping the \( w \) w. r. t. number of examples per category looks moderately lower for the LPS than for SIFT, indicating that using less examples with LPS decreases only the number of examples, we repeated 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), not being clearly in favor of any choice of parameters. This justifies using a shorter version of the descriptor, if performance speed (caused e. g. by a large number of regions used) is an issue. However, the dashed lines indicate that the larger descriptor is more stable for varying category weights. Even if this is no longer a parameter in an approximate search setup, we expect it to be beneficial to descriptor performance and thus chose to work with the larger, \( 10 \times 6 \) version.

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 (47\% in case the shorter descriptor is used). 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 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].

As the current LPS is only rotation and translation invariant, we are working on improving the scale-invariant LPS to reach the performance of the current descriptor. Further experiments have to be carried out to compare the performance of the scale-invariant and scale-variant LPS on a database with a focus on scale change. 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


