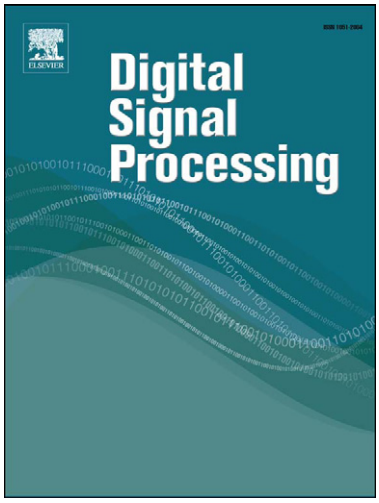


Component evolution analysis in descriptor graphs for descriptor ranking

Levente Kovács, Anita Keszler, Tamás Szirányi

PII: S1051-2004(14)00135-3
DOI: 10.1016/j.dsp.2014.04.013
Reference: YDSPR 1605

To appear in: *Digital Signal Processing*

Please cite this article in press as: L. Kovács et al., Component evolution analysis in descriptor graphs for descriptor ranking, *Digital Signal Process.* (2014), http://dx.doi.org/10.1016/j.dsp.2014.04.013

Highlights

Core findings of this work:

- We show that image/video datasets and descriptor performance can be efficiently represented by random geometric graph models.
- We show that analysing the phase transition of such graph models can be used for descriptor ranking.
- We present a ranking function for graph analysis that can be used for automatic feature selection and descriptor evaluation.
- Although the presented scheme is descriptor-independent, we evaluate and validate the approach on image/video datasets.
- The goal is to build an evaluation framework where descriptors can be analysed for automatic feature selection.

Component Evolution Analysis in Descriptor Graphs for Descriptor Ranking

Levente Kovács[1,*], Anita Keszler[1], Tamás Szirányi[1]

Distributed Events Analysis Research Laboratory, Institute for Computer Science and Control
Hungarian Academy of Sciences (MTA SZTAKI)
Kende u. 13-17, 1111 Budapest, Hungary

Abstract

This paper presents a method based on graph behaviour analysis for the evaluation of descriptor graphs (applied to image/video datasets) for descriptor performance analysis and ranking. Starting from the Erdős-Rényi model on uniform random graphs, the paper presents results of investigating random geometric graph behaviour in relation with the appearance of the giant component as a basis for ranking descriptors based on their clustering properties. We analyse the phase transition and the evolution of components in such graphs, and based on their behaviour, the corresponding descriptors are compared, ranked, and validated in retrieval tests. The goal is to build an evaluation framework where descriptors can be analysed for automatic feature selection.

Keywords: Descriptor Evaluation, Feature Extraction, Feature Selection, Graph Representation, Graph Components

1. Introduction

Content based retrieval in large video/image datasets is highly dependent on the choice of discriminating features and efficient index structures. Recent approaches involve graph clustering, clique searching, and component analysis methods. Open issues remain how to build the graphs (selection of edges and weights),

*Corresponding author
Email addresses: levente.kovacs@sztaki.mta.hu (Levente Kovács), keszler.anita@sztaki.mta.hu (Anita Keszler), sziranyi@sztaki.mta.hu (Tamás Szirányi)
[1]URL: http://web.eee.sztaki.hu

and how to navigate them efficiently (neighbourhood search). We propose and work towards proving that graph theoretic approaches can be useful in content based retrievals, for descriptor evaluation and automatic feature selection. We build our approach on the investigation of entity difference distributions according to several descriptors and analysing their relations and behaviour during component formulation and the appearance of the so called giant component in the graphs of the descriptors. As we will detail later, as the novelty of our approach, our goal is to exploit the inherent properties of the graph representations to evaluate descriptors based on the behaviour of their graphs during the formulation of the giant component, analysing their discrimination capabilities. The presented method has some connections to graph clustering methods in the sense that the effects of the descriptors on the structure of their graphs is related to their clustering properties.

When searching for similar content in video/image datasets, we need to apply feature extractors that gather information about the content and structure of the stored data, and use that information to create a searchable index for the dataset, which in turn will be the basis of searching for similar content. However, there are a lot of different descriptors, and usually it is very hard to select those, which perform well for a given dataset, when using them to produce retrieval results. Our goal is to help this process by providing a means to evaluate a set of descriptors for a given set of classes and data, and to find a combination of descriptors that perform better. This information can then be used to create more efficient indexes and produce higher precision retrievals.

Feature selection in the presence of irrelevant features (noise) is presented in [1], taking into consideration sample data points in 2D for boundary selection and investigating the distribution of feature weights in high dimensions. A method for feature selection [2] is based on approximately 1000 features on real videos, using heuristics for feature retention, using the sort-merge approach for selecting ranked feature groups. A method for sport video feature selection is presented in [3]; [4] presents a method for automatic image annotation based on a feature weighting scheme and machine learning; [5, 6] present similar approaches for feature selection based on mutual information and principal component analysis. [7] presents a query by example approach where histograms of point distances are investigated as a basis to show that with increased dimensions the distance distributions tend to be narrower (poor discrimination), and SIFT feature distribution histograms are used to improve clustering and retrieval.

Graphs are a natural way of representing data structures, describing interconnections and internal structures of datasets, visualizing relations and distances

of elements, and finding subsets, clusters and communities in such structures. Graphs have been widely used for clustering applications, including spectral clustering [8] for graph partitioning, MST (minimum spanning tree) based clustering [9], dense sub-graph mining [10], etc. The uses of graph clustering approaches are various, from generic pattern recognition (e.g., [11, 12]), to the recently highly researched community detection approaches in graphs representing social structures [13, 14, 15, 16].

Contrary to other approaches, we do not use artificial feature weighting or a priori clustering, instead we use real data with multiple features and weigh the built graphs by the points' differences according to features, and investigate the behaviour of the distributions. The goal is to show that this method is a good alternative to previous ones for finding features with higher discriminative properties. In our earlier work [17] we have proposed the use of descriptor graphs for descriptor ranking, and we produced a fitness function for providing such a rank [18]. This work extends these previous results by deeper investigation of the properties of such graph structures, regarding similarity in behaviour and topography, and the use of such intrinsic properties for feature selection.

We will start by introducing basic concepts and random geometric graphs (Sec. 2), followed by the description of the proposed parameters for ranking based on phase transition and component behaviour of descriptor graphs (Sec. 3), then the presentation of the used datasets and descriptors (Sec. 4), and finally the presentation of the ranking function and the performed evaluations (Sec. 5).

2. Component analysis of random graph models

In this section we overview the properties of two frequently applied random graph models and their component structures. Based on the results corresponding to random graphs, we get a better understanding of the properties of real-world graph structures. Let us start with some definitions of important terminology.

Definition 1. *An undirected graph is a G=(V,E) pair, where V denotes the set of vertices (or nodes) and E denotes the set of edges. $E \subseteq V \times V$ is a symmetric binary relation on V. The edges represent connections between the vertices of the graph, $e_{ij} \in E$ being an edge connecting vertices v_i and v_j.*

Definition 2. *The neighbourhood of a vertex $v \in V$ is $N(v) = \{w : (v,w) \in E\}$. The degree $d(v)$ of a vertex v is the number of its neighbours.*

Definition 3. *Graph $G' = (V', E')$ is a sub-graph of G if $V' \subseteq V$, $E' \subseteq E$ and if $e_{ij} \in E'$ then $v_i, v_j \in V'$.*

Definition 4. *If $W : E \mapsto \mathbb{R}$ is a weight function on $G = (V, E)$, then we say that the graph is weighted and a w_{ij} weight value corresponds to an edge e_{ij}.*

Definition 5. *A $G = (V, E)$ graph is connected, if there is a path between any two vertices. A path is a sequence of vertices in the graph, where neighbouring vertices of the sequence are adjacent in the graph, and a vertex appears only once in the sequence.*

Definition 6. *C is a component of $G = (V, E)$, if C is a sub-graph of G and it is connected. The size of a component is the number of vertices it contains.*

Definition 7. *A random geometric graph (RGG) is obtained as follows. We pick n random node position values as $X_1, X_2, ..., X_n \in \mathbb{R}^d$ (according to a probability distribution ν on $\mathbb{R}^d$, where d is the number of dimensions). We connect two nodes v_i and v_j ($i \neq j$) if their distance $\| X_i - X_j \| < r_n$, the radius of the graph.*

The theory of random graphs has an important role in discrete mathematics since the early 60's. Besides the theoretically interesting problems, random graphs have proven to be useful in engineering applications as well. Although real-world datasets are usually too complex to mimic each of their properties with synthetic datasets, some important parameters of their structure can be exposed by analysing random graphs. Famous examples are social networks [13, 16] and web graph analysis [19, 20].

The network parameters frequently modelled by random graphs are: the probability of the existence of certain edges of the real graph, the degree constraints, or - in case of weighted graphs -, the weights' distribution. After the model is built, some structural patterns get revealed, such as the number or size of components, cliques, or the occurrence of some special sub-graphs.

In our case, random graphs are used to analyse the number and size of components in real graphs. We aim to compare graphs built from test datasets based on a well known phenomenon in random graphs, namely the appearance of the giant component (defined in Sec. 2.1, Theorem 1). Our test results provide evidence of the existence of a component in these graphs with similar behaviour to the giant component (GC) in random graphs. Besides the properties of the GC in real graphs, we are also interested in the size and number of the components as well.

2.1. The Erdős-Rényi model

Erdős and Rényi analysed the properties of random graphs with uniformly distributed edges [21]. They considered the evolution of components, while adding randomly selected edges to the graph. The process starts with n vertices and 0 edges, and in each step a randomly selected new edge is added, independently of the already chosen edges. After each step, the size and number of components are studied. During the evolution of the graph, connected components start to appear and, when reaching a critical point, they merge into a so called giant component (GC).

The Erdős-Rényi model (ER-model) was originally described by the number of vertices and edges at a given step of the evolution: $G(n,e)$, where n denotes the number of vertices, and e is the number of edges. Recent results connected to this problem are formulated using the number of vertices and the p probability of the existence of an edge $G(n,p)$. If the edges are selected independently, this formulation gives the same result (as the above (n,e) description), and p is usually described as a function of the number of vertices: $p = c/n$, where c is a constant. A complete graph with n vertices has $n(n-1)/2$ edges, that is a $G(n, p = 1/n)$ graph ($c = 1$) corresponds to the ER-model with $n/2$ edges.

One of the most interesting results of Erdős and Rényi is a theorem [21] that can be formulated as follows:

Theorem 1. *(Erdős-Rényi) The behaviour of the ER graph model can be divided into three important phases, from the point of view of component sizes (where the size of the largest component is denoted by C_{max}):*

1. *$c < 1$: $C_{max} = O(\ln n)$ (the graph contains only small components);*
2. *$c = 1$: $C_{max} = O(n^{2/3})$;*
3. *$c > 1$: $C_{max} = O(n)$ (the giant component appears), and all other components have size $O(\ln n)$.*

The results presented in [21] also deal with the complexity of the components, but now we are interested in their sizes. The important consequence of this theorem is that after a given number of edges, a unique giant component (GC) appears. Below this threshold all components are small - the probability of a component containing a large fraction of the vertices is 0. Above this threshold, the probability of the existence of a GC is 1.

This statistical model has a well known application in percolation theory. From the late 50's, the attention was drawn towards cluster size and percolation problems and their applications [22, 23]. Let us have an infinite graph of sites

(vertices) and bonds (edges), where particles occupy sites, and a site is occupied with a fixed probability p, independently from others. There exists a critical probability p_{crit}, so that if $p < p_{crit}$, all clusters formed by occupied sites will have a finite size, but if $p > p_{crit}$, a cluster with infinite size will appear. It has been proven that the phenomenon described by Erdős and Rényi is the same as this type of so called percolation problem.

Although numerous scientific results are connected to the appearance of the GC in the ER-model, its existence was investigated in more complex random graph models as well, but there are still several open questions.

2.2. Random geometric graphs

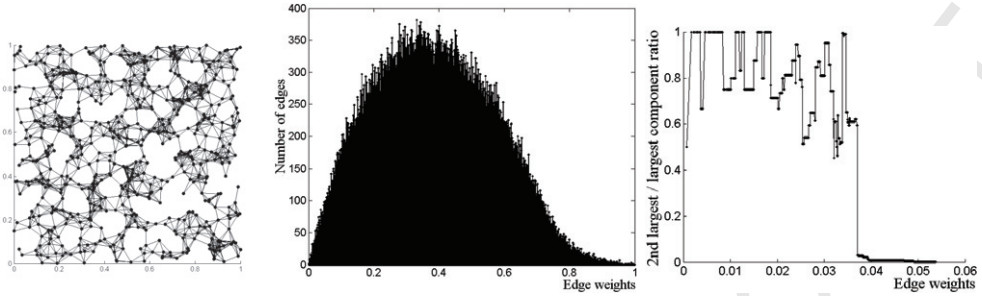
Besides the mentioned classical random graph models, several versions have been published for modelling certain properties of complex real networks. In our case, where the graph models images/videos and their distances, the most suitable model is the geometric graph. In this model, the edge weights correspond to pairwise distances of nodes based on a given metric.

The random geometric graph model (see Definition 7) offers a solution that mimics real network properties in a synthetic environment. The existence of the giant component (GC) in random geometric graphs has also been examined. A radius threshold is used to select graph edges: edges with a weight lower than the radius are selected [24, 25, 26]. The thermodynamic limit, a term from statistical physics, was also used to describe this phenomenon. This limit corresponds to the critical radius of the RGG: $r_n \sim c \cdot n^{-1/d}$. At this limit, the expected value of the average degree in the graph tends to a constant (c). Above a certain c, a GC is likely to appear. The existence probability and the uniqueness of this GC is the same as in the ER-model. Another important point worth mentioning is, that this model has a strong connection with Poisson processes as well [27, 28].

An example of a 2-dimensional RGG of 600 vertices is presented on Fig. 1 (a). Fig. 1 (b) shows the distribution of the edge weights, while (c) illustrates the ratio of the sizes of the second largest to the largest components while adding edges to the graph by their increasing weights (until the GC appears).

3. Graph structure analysis on real datasets

In this section we present the proposed graph analysis scheme based on investigating the component evolution process and the phase transition in descriptor graphs, and the proposed behaviour parameters that we will use in the ranking/fitness function for automatic descriptor ranking.



(a) Uniformly distributed RGG in 2D. (b) Histogram of normalized edge weights. (c) Ratio of the 2nd largest and largest component sizes.

Figure 1: Uniformly distributed 2D RGG of 600 vertices.

3.1. The appearance of the giant component in case of non-uniform weight distribution

The previously introduced examples (ER-model, RGG) are both statistical models. They have important roles in studying the behaviour of some network parameters such as the evolution of components in the graph. However, from several points of view, these models are only simplifications of real networks. In this section, we will present some of the issues of analysing real-world datasets.

The appearance of the GC in real networks with geometric restrictions on the edge weights is an interesting mathematical topic of its own. The number of vertices and the number of dimensions are relevant parameters to determine the circumstances of the appearance of the GC in an RGG. It is essential to note that the definition of the RGG contains a restriction on the distribution of vertex coordinates, i.e., the coordinates are uniformly distributed in each dimension. This restriction might be acceptable in certain applications such as sensor networks that can be modelled by a low dimensional (2D or 3D) geometric graph, but in most application areas, the positions of the vertices are not that structured.

In case of image/video datasets, the vertices are placed in a descriptor space with significantly larger number of dimensions, and we have no a priori knowledge on the distribution of coordinates. As it was mentioned in Sec. 2.2, even in the case of uniform distribution, the exact place of appearance of the GC is unknown if the number of dimensions is high. There have been very few results published on the number and size of components in case of non-uniform weight distributions. To the best of our knowledge, the existence and the circumstances of the appearance of the GC in arbitrary geometric graphs are still open questions.

We analyse descriptor graphs to find out whether the appearance of the GC is traceable, and if it is, whether it is descriptor-dependent. Our objective is not the solution of the above mentioned general theoretical questions, but to study the importance of the component sizes in real world applications in case of arbitrarily structured graphs.

The behaviour of the descriptor graphs during the process of GC formulation is an indirect way of analysing the clustering properties of a descriptor. Appearing components are basically clusters of nodes that are close together according to the used metric. How these components evolve (appear and merge), is an indicator of the performance of the descriptor. Additionally, how and when the GC appears (visualized by phase transition and component evolution graphs) is in direct connection with the descriptor's performance. E.g., if the GC appears early at a point where we only have very small or very few components, then the used descriptor is probably not a good choice for describing the dataset (low discriminative properties). Also, we expect that descriptors with similar phase transition and component evolution will perform similarly in a retrieval process.

3.2. Descriptor graphs

In order to prove our assumptions, besides the place (critical weight) of the phase transition we also investigate the components' evolution process with regard to the applied weight thresholds. Fig. 2 shows the behaviour of a descriptor graph in a 3D visualization, where the axes are the edge weight thresholds (th), the sizes of existing components at a given threshold (comp.sizes), and the number of such sized components at a given threshold (no.comp.). We are looking for locations where there are sudden jumps in component sizes during the increase of the weight threshold (i.e., when smaller components suddenly merge into a large one), as an indicator of the estimated place of the appearance of the GC. This critical weight is a main area of interest in these graphs - such a region is shown with the big black arrow in Fig. 2 (a). For the example in Fig. 2 (a), Figs. 2 (b-c) show the visual structure of the descriptor graph right before (b) and at (c) the appearance of the GC (visualized with Gephi [29]).

Similarly to random geometric graphs, we build the descriptor graphs by using the dataset elements as nodes, and the distance between them - according to a selected descriptor - as edge weights. Thus, we will have a complete (i.e., fully connected) graph for each descriptor. Then, we use these graphs to evaluate the performance of the corresponding descriptor by analysing the components. The main steps of our approach are the following:

1. Calculate the $d(v_i, v_j)$ distances of all v_i, v_j node pairs for all descriptors d_k.

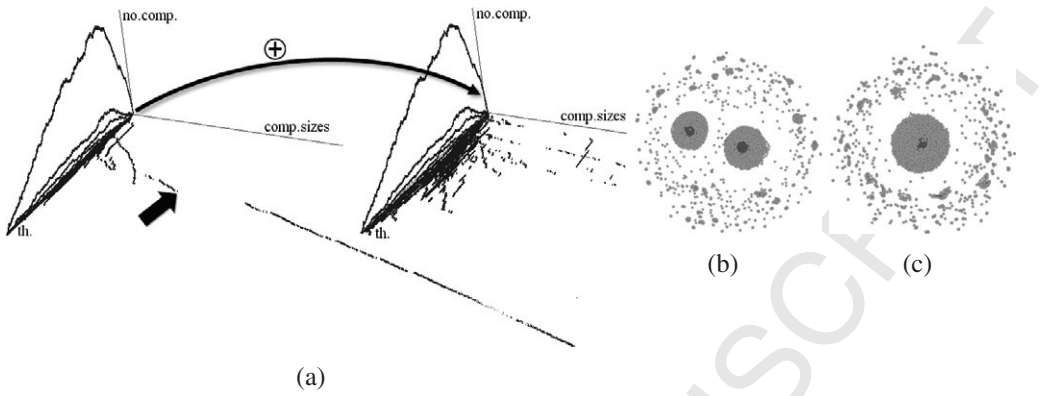


Figure 2: (a) 3D visualization of the evolution of graphs for homogenous texture descriptor according to changing weight thresholds (th), existing component sizes at a given threshold (comp.size) and the number of such sized components (no.comp.). Small black arrow shows the area of interest for the critical point where the last two largest clusters merge. The area near the origin is magnified (shown with the large arrow with a $\oplus$ sign) for a better view. (b) Component structure right before, and (c) at the critical point shown in (a).

2. For each descriptor d_k,
   (a) Start building the graph, $G(V,E)$, with vertex set $V = \{v_i | i = 1...n \in \mathbb{N}\}$ and weighted edge set $E = \{e_k | e_k = (v_i, v_j), i \neq j, v_i \in V, v_j \in V\}$, $|E| \leqslant n \cdot (n-1)/2$, with weights $w(e_k) = d(v_i, v_j)$ by gradually increasing the weight threshold $th \in [0,1] \cap \mathbb{R}$ and including all edges whose weights $w(e_k) \leqslant th$.
   (b) Merge components that have become connected by inserted edges, and iterate steps (a)-(b).
3. During the iterations in steps 2 (a)-(b), monitor the number and size of components (phase transition and component formulation) to find the appearance of the giant component (GC). Calculate parameters that describe this process and can be used to rank the descriptors.

Although the place of the appearance of the GC can not be exactly determined, the tests prove the existence of a single dominant component during the edge addition process. The parameter to estimate this critical threshold will be the ratio of the sizes of the second largest to the largest component. The estimation will be the weight threshold where this ratio decreases below 0.1, and will not exceed that threshold later.

Figure 3 shows, as an example, visual snapshots during the component be-

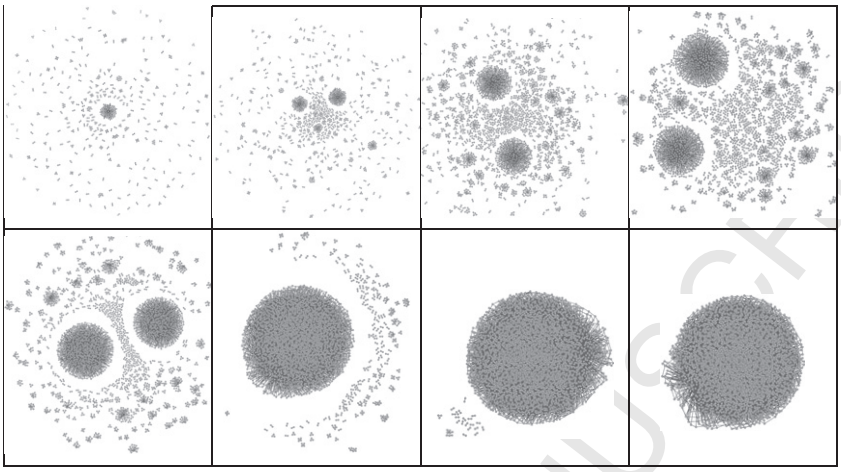


Figure 3: Visualization snapshots of component behaviour during the processing of the edge histogram descriptor (left to right, top to bottom). At early steps, there are many small components, then they grow and merge while raising the edge thresholds, until the GC appears.

haviour evaluation of the edge histogram descriptor. These visualizations have been produced by taking the component statistics from certain steps of the process and producing connected groups of vertices. Here, the locations of the vertices have no real meaning, the point is to visualize the number and sizes of components as they appear during the process. Fig. 14 shows visual excerpts from components during the descriptor graph building process, elements from the same components grouped together.

3.3. Component parameters in descriptor graphs

Above, we presented the graph building process. However, in case of larger datasets, we modify the process to gain a less detailed analysis, while keeping the significant steps. Instead of selecting single edges, we add all edges with the same weight in one step, and the critical edge number threshold is replaced by the critical edge weight threshold.

Test results of the critical threshold estimation are presented in Fig. 4. The size of the largest component of the graph (normalized with the number of vertices to show a common scale) is tracked during the building process of the descriptor graphs and its evolution is presented versus the increasing weight thresholds in the phase transition graphs of Fig. 4 (a),(c). The largest component grows rapidly within a small weight range in case of all descriptors, which is an expected behaviour based on the general theoretical knowledge regarding the evolution of

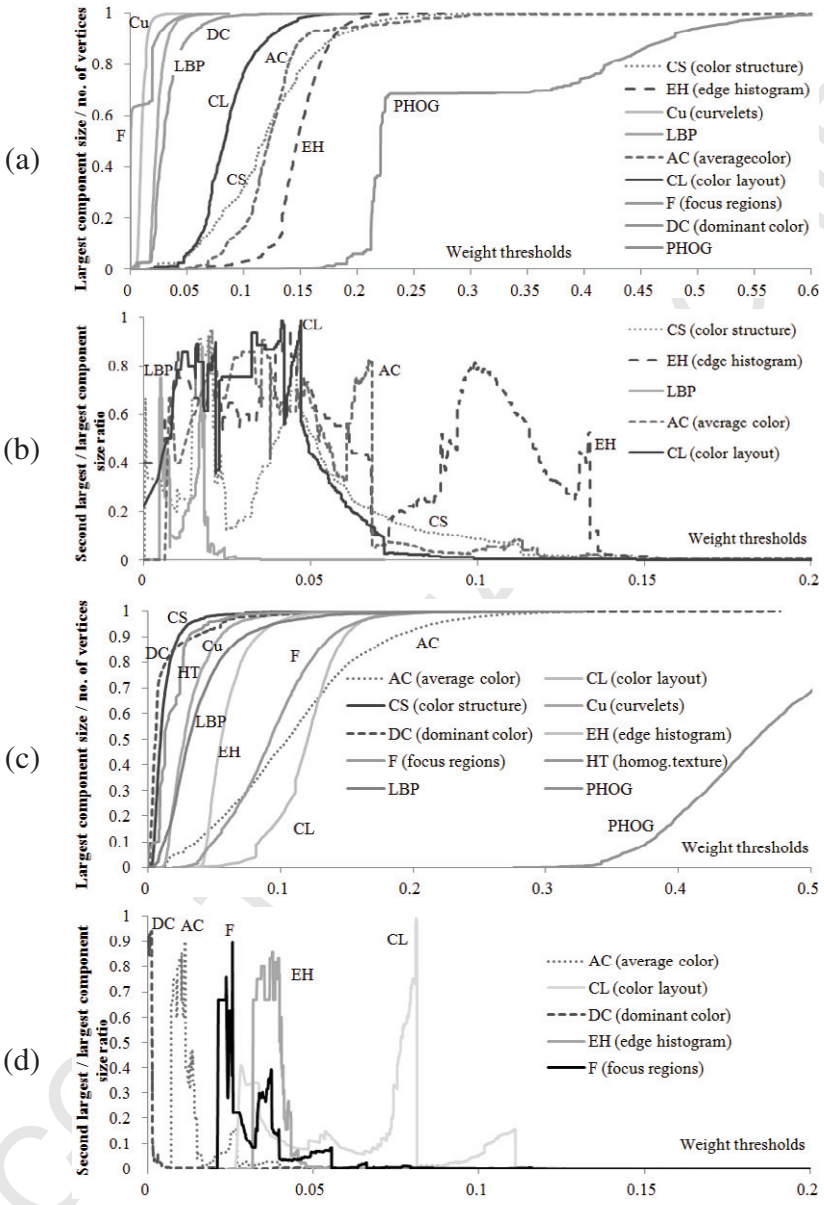
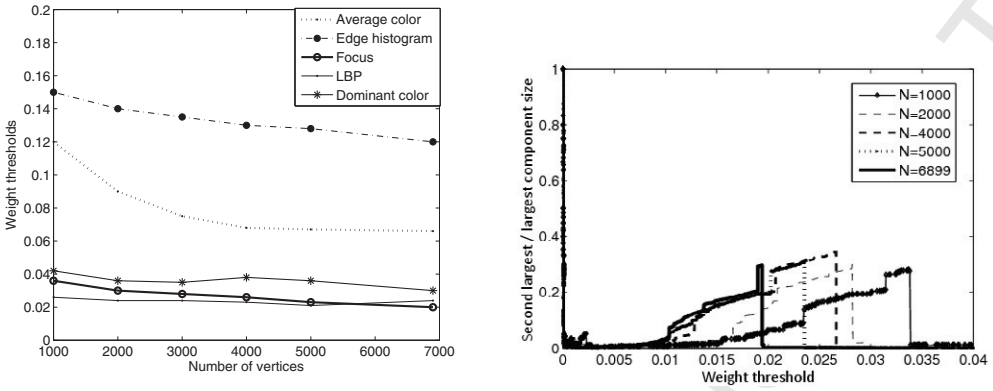


Figure 4: Phase transition graphs for several descriptors (weight thresholds vs. normalized largest component size) for the CDB7k (a) and the MIRFLICKR25k (c) datasets. Second largest to largest component size ratios vs. changing weight thresholds for some descriptors for CDB7k (b) and MIRFLICKR25k (d). The datasets are described in Sec. 4.



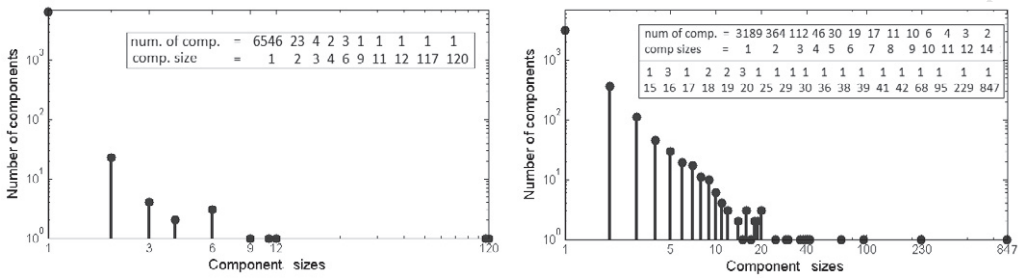
(a) (b)

Figure 5: (a) Critical weight values in the graph of different descriptors. (b) Ratio of the second largest to the largest component sizes of the focus region descriptor graph (N is the number of used dataset elements).

RGGs and the existence of a GC. Although this parameter is insufficient on its own to distinguish between descriptors - e.g., the evolution of the largest components of some descriptors might be similar -, the functions clearly shows the sudden emergence of a large component (note the steepness of the curves in a short weight range). However, the GC phenomenon also means that this large component becomes unique. To prove that the components of the descriptor graphs meet this requirement, we also observe the changes in the ratio of the sizes of the second largest to the largest components (Fig. 4 (b),(d)). Using both these parameters, most descriptors will show significant differences. For space considerations, a selected number of descriptors for two of the used datasets are displayed in Fig. 4, so as differences in behaviour can be visually observed, showing by example that indeed the phase transition curves of different descriptors show discriminable differences. However, since the presented method is descriptor-independent, the selection of the descriptors does not really matter, only that their differences can be observed and determined.

Critical edge weights for some of the descriptor graphs on the CDB7k dataset (for details see Sec. 4) are shown in Fig. 5 (a). As it shows, the critical weights depend on the number of vertices of the graph, but the impact of this parameter depends on the descriptor. Detailed behaviour of the critical weight values of the focus region descriptor over the CDB7k dataset are shown in Fig.5 (b).

However, there is a third parameter that should be considered: the number of



(a) Average colour descriptor (b) Edge histogram descriptor

Figure 6: Component sizes before the critical threshold for different descriptors.

components near the critical threshold where the GC suppresses the others (e.g., Fig. 6). The edge histogram-based graph near this weight value consists of more components than the one corresponding to the average colour descriptor, which means that before the largest component becomes dominant, it performs better in dividing the vertices. This is exactly the property that can serve as the base for automatic selection of better discriminating descriptors.

To conclude the above discussed graph properties, we summarize the interesting parameters of descriptor graphs that we use to produce the automatic descriptor ranking:

- w_{crit}, the estimated critical weight where the GC appears, expecting that performance should be better if the GC appears at a later stage,
- $|C_2|/|C_{max}|$, the ratio of the second largest to the largest component sizes at the appearance of the GC. A high ratio means that the GC was formed from larger components, a lower one means a high number of smaller components existed before the GC. This ratio is a direct controller of the graph building process, stopping when it reaches below a specified threshold (typically 10%). Graphs showing the behaviour of this ratio for some descriptors are shown in Fig. 4 (b), (d).
- $nrcomp/n$, the number of components (normalized with the number of vertices n) at the appearance of the GC, relating to how the nodes have been encompassed in components.

Figure 7: Example representative frames from the CDB7k dataset.

4. Datasets and descriptors

For evaluating the viability of the proposed approach, we use various datasets. One is the publicly available MIRFLICKR25000 dataset [2] [30] (denoted by MIRFLICKR25k), which contains 25000 images gathered from Flickr, along with tags, and is roughly partitioned into 24 categories with lots of overlaps. Another dataset we used is our own video and image dataset (denoted by CDB7k), which contains approximately 7000 video segments (515 minutes total length) collected from television captures in 13 categories, e.g., sports, nature, cartoons, music, cooking, news, street surveillance, outdoor, indoor (some examples in Fig. 7). The videos were automatically cut into shots and manually labelled into categories. For each shot a representative frame was automatically extracted. Video features are extracted from the shots, while image features from the representative frames of the shots. Another dataset is the publicly available University of Washington (UW) dataset [3], which contains 1333 images in 22 categories with annotations. The last dataset is the INRIA Holidays (IH) dataset [31] [4], which contains 1491 images in 500 categories.

We extracted all the features for images and video segments for all dataset elements from CDB7k, WU and IH. In the case of the MIRFLICKR25k dataset, we use 8000 elements for descriptor evaluation and ranking, and use the full dataset for evaluating retrieval performance.

For evaluating features for general distribution and content differentiation, we selected a set of various descriptors, namely: average colour (custom descriptor

[2] Detailed description can also be found at `http://press.liacs.nl/mirflickr/`

[3] `http://www.cs.washington.edu/research/imagedatabase`

[4] `http://lear.inrialpes.fr/~jegou/data.php`

that produces average representative colour values over image areas, denoted by *d1*), curvelets [32] (*d4*), focus regions [33] (*d7*), local binary patterns (LBP) [34] (*d9*), MPEG-7 descriptors [35] - colour layout (*d2*), colour structure (*d3*), dominant colour (*d5*), edge histogram (*d6*), homogeneous texture (*d8*), motion activity (*d11*) -, average motion (custom descriptor gathering average representative motion directions over frame areas, denoted by *d12*), PHOG - pyramid of histograms of orientation gradients [36] (*d10*). However, there is no limit on the number of descriptors that could evaluated in our framework. For calculating the difference between images/videos we used Euclidean distance calculations, i.e., for a feature all elements can be displayed along a 1D axis from 0 to $d_{max}(D)$ (maximal difference for a descriptor) and they adhere to the triangle inequality.

5. Descriptor ranking and evaluation

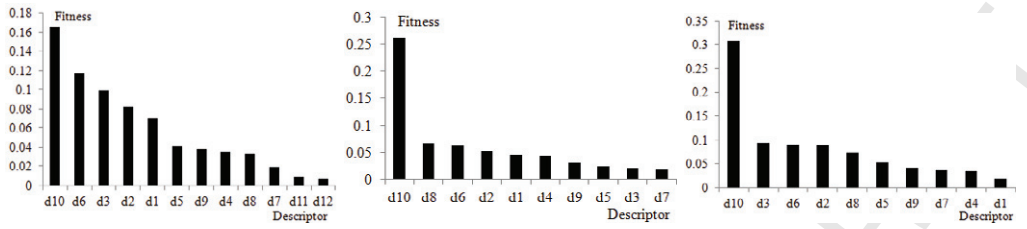
We discussed the important parameters of the components of the descriptor graphs in Sec. 3.3. Here, we present the suggested ranking function based on these parameters. The parameters can be weighted depending on the dataset, the number of classes we have, and the level of classification we target. In [18], we introduced a fitness/ranking function that combines these parameters into a formula:

$$F(\cdot) = w_1 \cdot w_{crit} + w_2 \cdot |C_2|/|C_{max}| + w_3 \cdot nrcomp/n \,, \tag{1}$$

where $w_{1,2,3}$ are weights that are manually specified and are constant for a given dataset (we intend to work on creating a process for adaptive weight selection in the future). Further on, we use this formula to produce a descriptor ranking and perform the evaluations, with weights always (for all cases and all datasets) set to $0.7, 0.2, 0.1$, respectively, giving higher importance to the critical weight where the GC appears, but also taking into account the other parameters, the importance of which was discussed in Sec. 3.3.

It is important to note, that the calculation of the above function does not depend on the used dataset, or the used descriptors, but only on the structure of the analysed graphs, and can be calculated without an a priori training or labelling process. In our experiments, we only used the class labels in the datasets to evaluate the performances of the retrievals. However, the described descriptor ranking process could be also used to select better performing descriptors not only on a 'blind' dataset, but on selected sub-categories as well, which could help in establishing descriptor pools for any particular content (sub-)class.

We calculated the F fitness values for the participating descriptors to produce a ranking based on the graph analysis results, taking into consideration the GC



(a) (b) (c)

Figure 8: Descriptor ranking produced by the used fitness function for (a) CDB7k, (b) MIR-FLICKR25k, (c) UW. Since the CDB7k is a video dataset, the motion-based descriptors are also used (d11 and d12).

appearances in the respective descriptor graphs. Such ranking results are shown in Fig. 8 for the used datasets.

In [18] we have shown that the produced descriptor ranking based on the above fitness function produces a ranking close to the exhaustive ground truth ranking, while at the same time has important benefits:

- it does not require exhaustive evaluation for all categories and all descriptors in the dataset to produce a ranking,
- it can produce a descriptor ranking for a given dataset while the method itself is independent of the number of categories and descriptors, providing a ranking based on the discriminating properties of the descriptors.

We evaluate the proposed framework by running retrievals on the used datasets. As a baseline, we use our parallel multi-tree indexing scheme [37] with and without exploiting the obtained ranking/fitness information, and compare the results. This retrieval uses a scheme in which the search is done for each included feature in parallel and the results are combined in a form of result aggregation process. However, any other similar indexing-retrieval scheme could be used just as well.

Overall, retrievals using the produced descriptor ranking can produce results that have similar or higher precision using a reduced number of descriptors, and the results contain lower variation in the number of categories that are irrelevant (not belonging to the query's category). In practice this means that in the case of rank-based retrievals the responses contain more relevant results, essentially decreasing the 'noise' of the retrievals (showcased by Figs. 9 and 15).

The above properties are shown in Fig. 9 for the CDB7k dataset, where 4 queries are used with a fixed result retrieval number of 50 and 100 (the first 50 and 100 best matches are returned), and using two retrieval approaches: $v1$, where

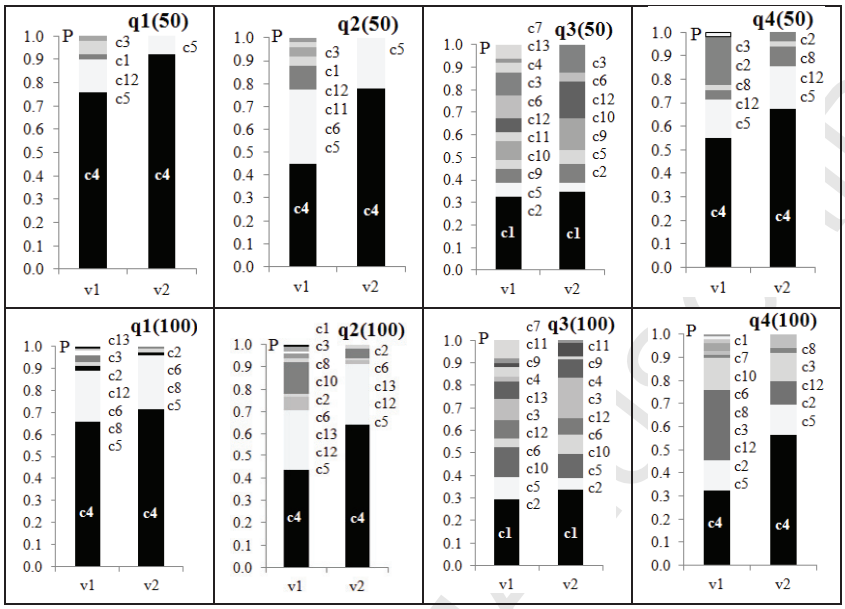


Figure 9: Retrieval results (precision) for 4 queries (q1-4) for first 50 and first 100 best results (qi(50) and qi(100), $i = \overline{1,4}$). In all graphs *v1* columns represent retrievals without descriptor ranking, while *v2* columns represent retrievals where descriptor ranking is taken into consideration. The 'cX' notations beside the columns show the class labels that make up the irrelevant results (fewer in case of *v*2 retrievals).

results are retrieved without the produced descriptor ranking information (results are generated using all available descriptors), and *v*2, where results are retrieved using the produced ranking (results are generated using the first 7 best performing descriptors). In the case of *v*2 retrievals, there might not just be an improvement in precision (the results are more relevant), but the remainders of the results show less variation (i.e., results were generated by descriptors with better clustering properties). This is shown by 'cX' labels along the columns, representing the labels of classes that make up the irrelevant results. Fig. 15 shows visual results for the two types of retrievals, showing the difference between the responses, where *v*2 (with rank) was only 16% better, and the images in the shown sample all belong to the same class ("sport"), however, the visual consistency of retrievals based on descriptor ranking is much higher.

For evaluating the eventual (expected) improvement in retrieval precisions (ratio of relevant results vs. all retrieved), we ran several queries, retrieved the closest 50 data points, and calculated the retrievals' precision. Fig. 10 shows results on the used datasets. Here, *v*1 again refers to retrievals without the use of ranking

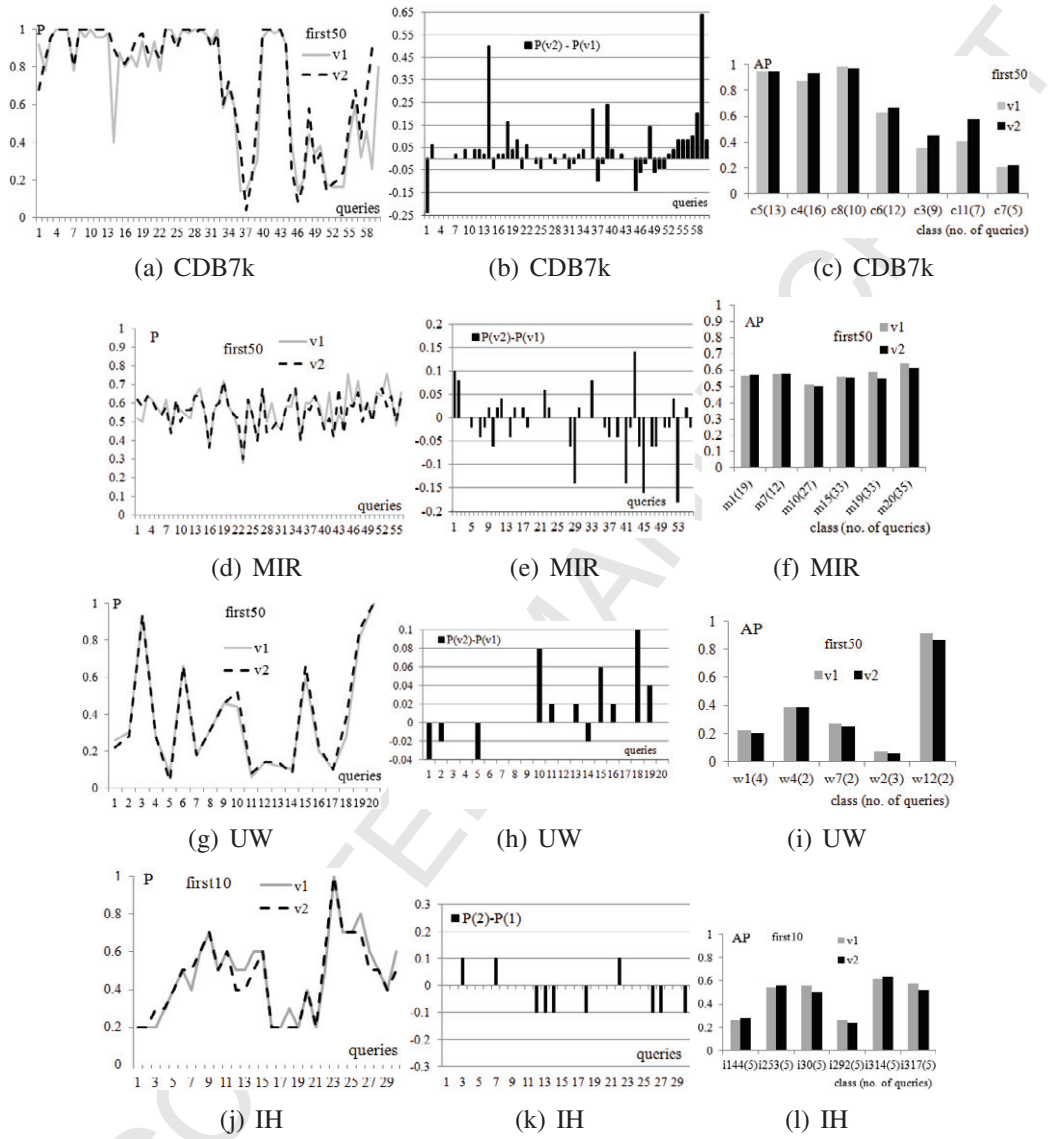
information (all descriptors are used), and $v2$ refers to retrievals which use the ranking data, only using the top 7 better performing descriptors in the retrieval process. In general, we can say that ranking information preserves or improves precision, Figs. 10 (b), (e), (h), (k) show the differences in precision values. For CDB7k, in 52% of the retrievals the precision improved and in 22% it remained the same, producing on average increase of 10.2%, while in the 26% of the cases where the precision was lower, the average decrease was only 5.8%. Fig. 10 (c) shows average precisions (AP) calculated for the tests in (a) with queries belonging to the same class grouped and averaged. For UW (g-i), the precision was better or equal in 80% with an average increase of 5%.

Fig. 10 (d-f) shows the precision values for 56 queries over the MIRFLICKR25k dataset, with one important note: we only used 8000 of the dataset's 25000 elements to create the descriptor ranking, but used the whole dataset in the retrieval process. Even in this case, the $v2$ results are better or similar in 62.5% of the retrievals. This in turn means, that the presented method is applicable to large datasets, by using a smaller sub-set to create the ranking.

Fig. 10 also shows another important aspect and goal of the proposed method: that, when using ranking information, we can achieve similar of better performance in the retrievals, but with a reduced set of descriptors, which in practice translates into a more streamlined and lightweight retrieval scheme.

The visual retrieval example (based on the CDB7k dataset) in Fig. 15 (a)-(b) shows that while the precision of $v2$ (with rank) was only 16% better, the visual consistency of the retrievals based on descriptor ranking is better (the used query is shown as the top-left image in both cases).

We also performed tests to see whether descriptors with similar properties behave similarly during a retrieval process (e.g., produce similar precision retrievals). This is an important issue when designing the collection of descriptors to use for a particular dataset, or when designing bag of words (BOW) retrievals, since if we know that some descriptors behave in a similar manner, then we do not have to use more or all of them, but we can select the best performing one. This in turn can make the retrieval process more lightweight. We selected 10 random queries from 5 different classes, and ran retrievals using only one selected descriptor at a time. Fig. 11 shows average precision (AP) values of these test runs (averaging 10 retrievals for each descriptor, using the CDB7k dataset as an example), where the first group of descriptors was selected from the middle of the phase transition graph (Fig. 4 (a)), and the other group from the left region, showing distinguishable difference between the two groups. This supports the expectation that phase transition properties are in correlation with internal topo-



(a) CDB7k (b) CDB7k (c) CDB7k

(d) MIR (e) MIR (f) MIR

(g) UW (h) UW (i) UW

(j) IH (k) IH (l) IH

Figure 10: In all cases $v1$ and $v2$ denote queries without and with ranking information (only using the 7 best descriptors), respectively. CDB7k dataset: (a) Precision (P) values for 60 queries. (b) Gain of $v2$ retrievals with ranking information. (c) Precisions averaged (AP) over queries from the same class. MIRFLICKR25k dataset: (d) Precision for 56 queries. (e) $v1$-$v2$ precision differences. (f) Average precisions. UW dataset: (g) Precision values for 20 queries. (h) Gain of $v2$ retrievals. (i) Average precisions. (j-l) Same for the IH dataset.

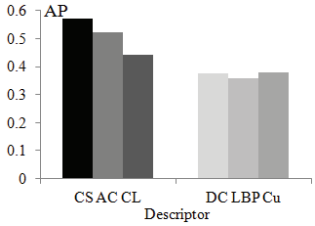
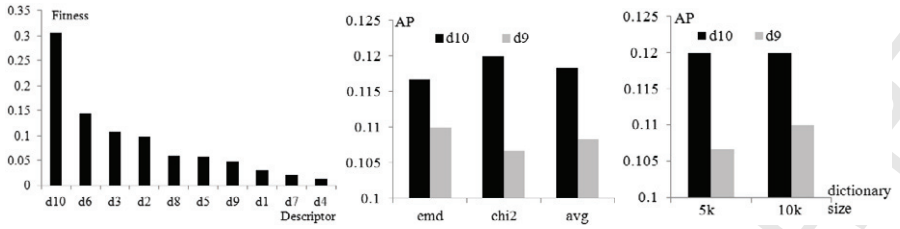


Figure 11: Average precisions (AP) for descriptors analysed in Fig. 4 (a) (for the CDB7k dataset) from the middle (CS, AC, CL) and left (DC, LBP, Cu) regions of the figure.

logical parameters of the descriptor graphs, which could be a topic of continued research in the area.

In order to show on a proof-of-concept level, that the proposed method could also be usable in other types of retrieval frameworks, we evaluated the performance of ranked descriptors in a bag of visual words type of retrieval test. Based on the IH dataset used earlier, we created a bare-bones retrieval using descriptors *d10* (PHOG) and *d9* (LBP), which were ranked better and worse, respectively (for the IH dataset, see Fig. 12 (a)). We tested the retrieval using 5000 and 10000 visual words, built separately for the two descriptors. We extracted feature points from the images, and used a fixed 32×32 region size around each point to extract the descriptors. For comparing the resulting visual word histograms during the retrieval, we used two standard metrics: the earth mover's distance (emd) and the χ^2 metric (chi2). We ran the same 30 retrievals as for Fig. 10 (j-l), and we expected that retrievals using the *d10* descriptor would perform better than using *d9*. Results are shown in Figs. 12 (b-c). Fig. 12 (b) shows the average precision (AP) values for the retrievals, using 5000 visual words and the two descriptors (*d10*, *d9*), the two metrics (emd, chi2) and their averages (avg). Fig. 12 (c) shows results for the same dataset, descriptors and metrics, comparing the results of using visual dictionaries of 5000 (5k) and 10000 (10k) size. The results also support the expected behaviour: ranking relates to performance.

Regarding the computational complexity of the proposed descriptor graph analysis approach, Fig. 13 shows normalized (by the number of nodes) computational times for all used datasets. In the worst case, the computation time is proportional to n^2 (n being the number of vertices); however, this can be reduced by more optimized coding, which was not our primary goal here.



(a) (b) (c)

Figure 12: Proof-of-concept results for bag of words retrieval using descriptors *d10* (ranked higher) and *d9* on the IH dataset (a). (b) Average precision (AP) for 30 queries, using 5000 visual words, two metrics (emd, chi2) and their averages (avg). (c) AP values for the two descriptors on different dictionary sizes (5k, 10k). The higher ranked *d10* shows better performance.

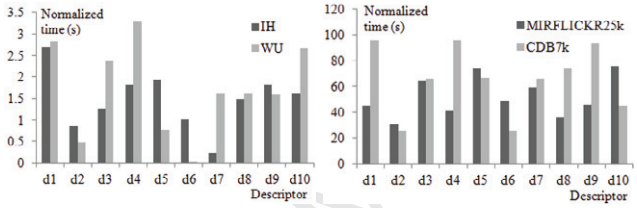
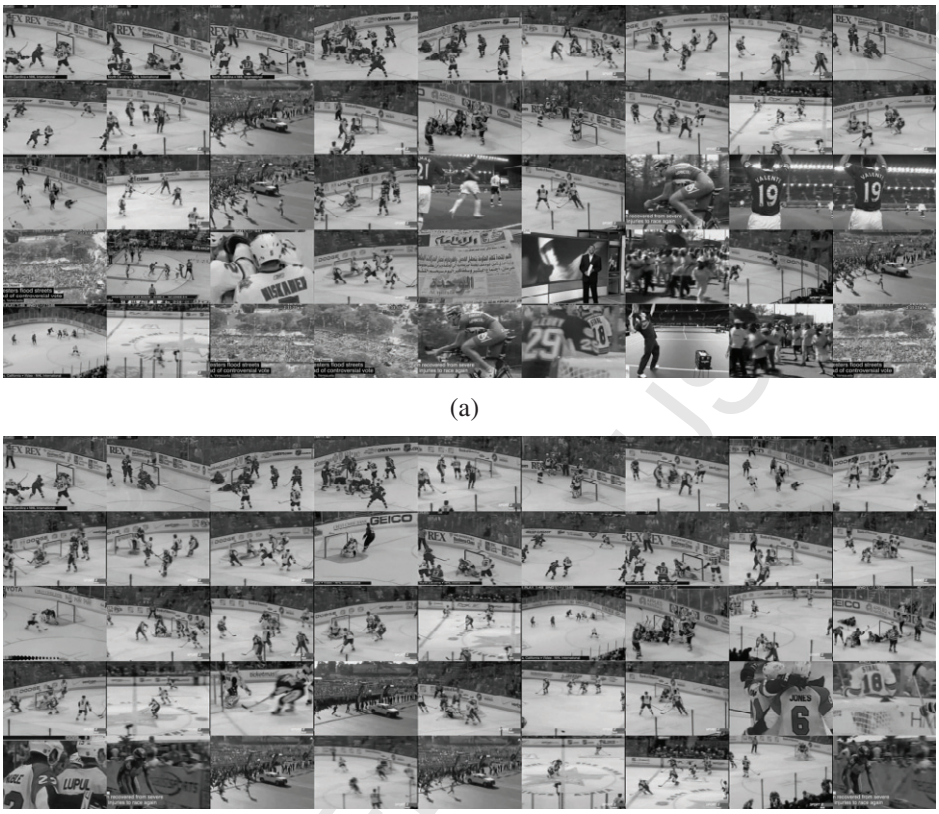


Figure 13: Normalized descriptor graph analysis computation times for the 4 used datasets.

(a) (b)

Figure 14: Examples of components at an inner building step of the (a) edge histogram (using CDB7k) and (b) texture graphs (using UW).



(a)

(b)

Figure 15: Sample retrievals for $v1$ (a) without rank and $v2$ (b) with rank information, showing an example where although the displayed items are all parts of the "sport" category, the $v2$ retrieval produces more visually consistent results. The query image is shown at the top-left of both (a) and (b).

6. Conclusions

We proposed a data- and descriptor-independent evaluation framework for descriptor evaluation and feature selection, exploiting descriptor graph behaviour analysis results with regard to the giant component formulation process. Any kind of descriptors can be evaluated this way, by providing their feature extraction algorithm and a metric to compare the features, and the datasets also can be other than visual (images/videos). The goal is to provide a means to select better performing descriptors for a given dataset, reducing the number of necessary descriptors, and producing more relevant results. We have presented relevant pa-

rameters and proposed a fitness function to rank descriptors. We experimentally showed, that the observation of the phase transition process is an effective and usable method for descriptor characterization. We performed evaluations to explore the practical viability of the suggested approach. As a practical continuation, we plan to apply the method to other large datasets, with the ranking step performed on a subset of content classes. We also intend to work on making the weighting of the ranking parameters adaptive, to further improve the descriptor discrimination and ranking results. In its current form, the proposed descriptor ranking method is suitable for automatic feature selection in image and video datasets.

Acknowledgements

This work was supported by Hungarian Scientific Research Fund (OTKA) grants nr. 83438 and 106374.

Vitae

Levente Kovács: Senior research fellow at the Institute for Computer Science and Control of the Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary. MSc in IT (2002), PhD in image processing and graphics (2007) from the University of Pannonia, Hungary. Main research areas: image/video feature extraction&indexing, annotation, event detection, non-photorealistic rendering, video restoration.

Anita Keszler: Research associate at the Distributed Events Analysis Research Laboratory, Institute for Computer Science and Control of the Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary. PhD student at the Faculty of Electrical Engineering and Informatics, Budapest University of Technology. Main research areas: graph based pattern recognition, MST-based clustering, pattern recognition in bipartite graphs, dense sub-graph mining, image segmentation, focused area selection, clustering image datasets.

Tamás Szirányi: Ph.D. (1991) and D.Sci. (2001) degrees by the Hungarian Academy of Sciences. Appointed as Full Professor in 2001 at University of Pannonia, Veszprém, Hungary, and at the Péter Pázmány Catholic University, Budapest in 2004. Leads the Distributed Events Analysis Research Laboratory at the Institute for Computer Science and Control, Budapest. Research areas include machine perception, stochastic optimization, remote sensing, surveillance systems for multi-view camera systems, intelligent networked sensor systems, graph based clustering, film restoration. Senior member of IEEE and a Fellow of the IAPR and the Hungarian Academy of Engineering.

References

[1] Y. Sun, S. Todorovic, S. Goodison, Local learning based feature selection for high dimensional data analysis, IEEE Trans. on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1610–1626.

[2] M. Morris, J. Kender, Sort-merge feature selection and fusion methods for classification of unstructured video, in: Proc. of IEEE International Conference on Multimedia and Expo, 2009, pp. 578–581.

[3] Y. Shen, H. Lu, X. Xue, A semi-automatic feature selecting method for sports video highlight annotation, in: Proc. of 9th Intl. Conference on Advances in Visual Information Systems, 2007, pp. 38–48.

[4] L. Setia, H. Burkhardt, Feature selection for automatic image annotation, in: Proc. of 28th Pattern Recognition Symposium of the German Association for Pattern Recognition, Springer, 2006, pp. 294–303.

[5] E. Guldogan, M. Gabbouj, Feature selection for content-based image retrieval, Signal, Image and Video Processing 2 (3) (2008) 241–250.

[6] F. Li, Q. Dai, W. Xu, Improved similarity-based online feature selection in region-based image retrieval, in: Proc. of IEEE Intl. Conference on Multimedia and Expo, 2006, pp. 349–352.

[7] W. Zhang, S. Men, L. Xu, B. Xu, Feature distribution based quick image retrieval, in: Proc. of Web Information Systems and Applications Conference, 2010, pp. 23–28.

[8] D. A. Spielman, Spectral graph theory and its applications, in: Proc. of 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2007, pp. 29–38.

[9] A. Keszler, T. Szirányi, Z. Tuza, Dense subgraph mining with a mixed graph model, Pattern Recognition Letters 34 (11) (2013) 1252–1262.

[10] V. E. Lee, N. Ruan, R. Jin, C. Aggarwal, A survey of algorithms for dense subgraph discovery, in: Managing and Mining Graph Data, Springer, 2010, pp. 303–336.

[11] H. Qiu, E. R. Hancock, Graph matching and clustering using spectral partitions, Pattern Recognition 39 (2006) 22–34.

[12] C. Y. Li, C. T. Hsu, Image retrieval with relevance feedback based on graph-theoretic region correspondence estimation, IEEE Trans. Multimedia 10 (2) (2008) 447–456.

[13] M. E. Newman, D. J. Watts, S. H. Strogatz, Random graph models of social networks, Proceedings of the National Academy of Sciences of the United States of America 99 (Suppl 1) (2002) 2566–2572.

[14] R. Albert, A. L. Barabási, Statistical mechanics of complex networks, Reviews of Modern Physics 74 (2002) 47–97.

[15] A. Keszler, T. Szirányi, A mixed graph model for community detection, International Journal of Intelligent Information and Database Systems 6 (5) (2012) 479–494.

[16] G. Robins, P. Pattison, Y. Kalish, D. Lusher, An introduction to exponential random graph (p*) models for social networks, Social Networks 29 (2) (2007) 173–191.

[17] A. Keszler, L. Kovács, T. Szirányi, Graph based descriptor evaluation for automatic feature selection, in: Proc. of Intl. Conf. on Computer Vision Theory and Applications (VISAPP), Vol. 1, 2012, pp. 375–380.

[18] A. Keszler, L. Kovács, T. Szirányi, The appearance of the giant component in descriptor graphs and its application for descriptor selection, in: Proc. of Conference and Labs of the Evaluation Forum (CLEF) Lecture Notes in Computer Science, Vol. 7488, Springer, 2012, pp. 76–81.

[19] B. Bollobás, O. Riordan, J. Spencer, G. Tusnády, The degree sequence of a scale-free random graph process, Random Structures and Algorithms 18 (2001) 279–290.

[20] A. L. Barabási, R. Albert, H. Jeong, Scale-free characteristics of random networks: The topology of the world wide web, Physica A 272 (2000) 173–187.

[21] P. Erdős, A. Rényi, On the evolution of random graphs, Publication of the Mathematical Institute of the Hungarian Academy of Sciences, 1960.

[22] C. Domb, M. Sykes, Cluster size in random mixtures and percolation processes, Physical Review 122 (1) (1961) 77–78.

[23] M. Fisher, Critical probabilities for cluster size and percolation problems, Journal of Mathematical Physics 2 (4) (1961) 620–627.

[24] M. Haenggi, J. G. Andrews, F. Baccelli, O. Dousse, M. Franceschetti, Stochastic geometry and random graphs for the analysis and design of wireless networks, IEEE J. on Selected Areas in Communication 27 (7) (2009) 1029–1046.

[25] G. Németh, G. Vattay, Giant clusters in random ad hoc networks, Physical Rev. E 67 (3) (2003) 036110.

[26] J. Dall, M. Christensen, Random geometric graphs, Phys. Rev. E Stat. Nonlin. Soft Matter Phys. 66 (1) (2002) 016121.

[27] R. Meester, R. Roy, Continuum Percolation, Cambridge University Press, 1996.

[28] M. Penrose, Random Geometric Graphs, Oxford University Press, 2003.

[29] M. Bastian, S. Heymann, M. Jacomy, Gephi: An open source software for exploring and manipulating networks, in: Proc. of Intl. AAAI Conference on Weblogs and Social Media, 2009, pp. 361–362.

[30] M. J. Huiskes, M. S. Lew, The MIR Flickr retrieval evaluation, in: Proc. of ACM International Conference on Multimedia Information Retrieval (MIR), ACM, 2008, pp. 39–43.

[31] H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometry consistency for large scale image searc, in: Proc. of the 10th European Conference on Computer Vision, 2008, pp. 304–317.

[32] E. Candes, L. Demanet, D. Donoho, L. Ying, Fast discrete curvelet transforms, Multiscale Modeling and Simulation 5 (3) (2006) 861–899.

[33] L. Kovács, T. Szirányi, Focus area extraction by blind deconvolution for defining regions of interest, IEEE Trans. on Pattern Analysis and Machine Intelligence 29 (6) (2007) 1080–1085.

[34] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. on Pattern Analysis and Machine Intelligence 24 (7) (2002) 971–987.

[35] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, A. Yamada, Color and texture descriptors, IEEE Trans. on Circuits and Systems for Video Technology 2 (6) (2001) 703–715.

[36] A. Bosch, A. Zisserman, X. Munoz, Representing shape with a spatial pyramid kernel, in: Proc. of 6th ACM Intl. Conference on Image and Video Retrieval, Springer, 2007, pp. 401–408.

[37] L. Kovács, Parallel multi-tree indexing for evaluating large descriptor sets, in: Proc. of IEEE Intl. Workshop on Content-Based Multimedia Indexing, 2013, pp. 173–178.