Scene-based Image Retrieval by Transitive Matching *

Adrian Ulges
German Research Center for Artificial Intelligence (DFKI)
Trippstadter Str. 122
Kaiserslautern, Germany
adrian.ulges@dfki.de

Christian Schulze
German Research Center for Artificial Intelligence (DFKI)
Trippstadter Str. 122
Kaiserslautern, Germany
christian.schulze@dfki.de

ABSTRACT

We address scene-based image retrieval, the challenge of finding pictures taken at the same location as a given query image, whereas a key challenge lies in the fact that target images may show the same scene but different parts of it. To overcome this lack of direct correspondences with the query image, we study two strategies that exploit the structure of the targeted image collection: first, cluster matching, where pictures are grouped and retrieval is conducted on cluster level. Second, we propose a probabilistically motivated *shortest path* approach that determines retrieval scores based on the shortest path in a cost graph defined over the image collection. We evaluate both approaches on several datasets including indoor and outdoor locations, demonstrating that the accuracy of scene-based retrieval can be improved distinctly (by up to 40%), particularly by the shortest path approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Retrieval and Indexing

General Terms

Algorithms, Measurement, Experimentation

Keywords

content-based image retrieval, similarity search, scene-based retrieval, image clustering

1. INTRODUCTION

Similarity search is a frequently studied challenge in content-based image retrieval, where – given a query picture – visually similar images are to be retrieved from a dataset. Applications range from copyright preservation over mobile image retrieval and archive search to multimedia forensics. With this diversity of application scenarios, the criteria for what defines a "similar" image vary strongly: sometimes, we want to discover content that is a re-encoded or otherwise modified version of an original image or video scene, for example to protect the interests of a copyright holder. In other situations, similarity search is targeted at images showing the same *object* as the query image.

In this paper, we address another type of similarity: our goal is to retrieve images taken at the same location or *scene* (e.g., an indoor room or an outdoor place) as the query image. We will refer to this challenge as *scene-based retrieval*. Its applications are in organizing personal image collections (imagine finding holiday snapshots taken at the same spot) or in multimedia forensics, where police investigators are interested in finding images associated with the same crime scene (e.g., to uncover links between different criminal cases).

One common approach towards addressing similarity search is local feature matching, in which regions of interest in both the target image and the query image are detected and cor-

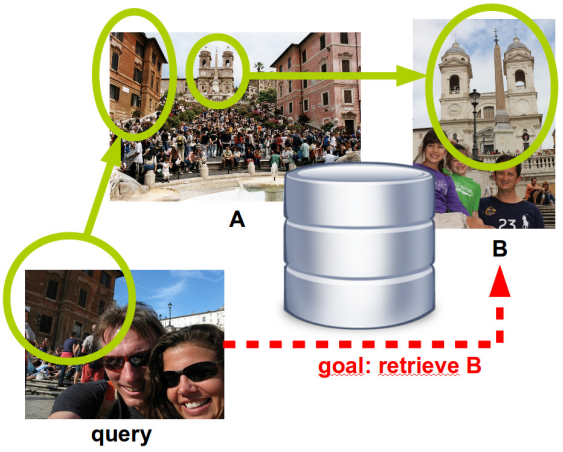


Figure 1: In scene-based image retrieval, target pictures (like B in this example) may show different parts of a scene, sharing no local correspondences with the query image. To retrieve them, we exploit the structure of the image collection using a transitive matching (here, the fact that image A is connected with B as well as the query image).

*

ICMR '11, April 17-20, Trento, Italy

respondences between the two feature sets are found. A fundamental challenge with scene-based retrieval is that target images may show a different part of the same scene, and thus not share any such correspondences with the query image. This illustrated in Figure 1, where image A shares some visual overlap with the query image but B does not.

To overcome this problem, we suggest to exploit the structure of image collections as an additional information source: our first strategy will be cluster matching, where we cluster the image collection and, – assuming that images in the resulting clusters will belong to the same scene – we retrieve images in order of the clusters they belong to. Second, we propose a probabilistically motivated *shortest path* approach, in which we formulate *costs* between images decreasing with their correspondence strength. Retrieval is conducted by finding images that can be reached via low-cost paths, which allows a transitive matching (for example, in Figure 1, a low-cost path from the query over A to B allows us to retrieve image B).

This paper is organized as follows: we first review related work (Section 2), followed by a description of both approaches (Section 3) and quantitative experiments on several datasets showing indoor and outdoor locations (Section 4), in which we validate strong improvements, particularly by shortest path retrieval. We finish with a conclusion (Section 5).

2. RELATED WORK

Patch-based Image Retrieval.

Since their introduction by Lowe [8], local features or patches have widely been applied to object recognition and other vision tasks. This local feature approach has also been used for content-based image retrieval [3, 21]. Many special purpose image retrieval systems have been developed for domains like medical, document and tattoo images [7, 19, 4]. An evaluation of different local feature descriptors [11] states that SIFT features outperform other local feature approaches regarding their matching quality. We relate to patch-based image retrieval due to the fact that we use SIFT features exclusively as our data source for the proposed approach. Here, we especially exploit the potential of local features for detecting the occurrence of multiple objects in different images in different constellations without the need of prior segmentation.

Stitching and 3D Reconstruction.

Photo Stitching and 3D reconstruction are related to our work, as they involve the linkage of unordered image sets via sparse correspondences. In this area, impressive results have been achieved recently [16] and 3D reconstructions have been generated from large-scale diverse image sets [2]. However, while a full 3D reconstruction works reliably on well-aligned images with strong correspondences, we focus on more complex situations, involving significant occlusion, object motion and articulation, and lighting changes. In these complex situations, it is difficult to decide whether two images do show parts of the same scene, and we present a light-weight robust solution for making this decision.

Object Recognition.

Object recognition is related to scene-based retrieval, as keypoints on both objects and scenes may be visible only from certain perspectives. Correspondingly, object views form topological structures (or *manifolds*) that have already been exploited as an information source [13]. Like object recognition, scene-based retrieval is targeted at associating test views with these manifolds. The main difference, however, is that object recognition is conducted in a supervised setting (i.e., training views are labeled), and the focus is usually on finding a *single* object view similar to the query view [13, 8]. In contrast, scene-based retrieval is targeted at discovering *all* views related to a scene, i.e. recover manifolds in an unsupervised fashion.

Image Clustering and Image Graphs.

Clustering has been used before as a strategy to exploit the internal structure of image collections: Tuytelaars et al. [20] evaluate a variety of clustering algorithms for an unsupervised mining for object categories. Other work has been targeted at finding outdoor locations: Quack et al. [17] perform a multi-modal clustering taking geo-tags, visual and textual clues into account. By linking clusters with wikipedia articles and matching images with clusters, auto-annotation is achieved. Philbin et al. [15] cluster large-scale image collections using a hybrid approach, which first oversegments the dataset and then recombines clusters of diverse views of the same object. We validate improvements by such cluster matching approaches for scene-based retrieval. Beyond this, we suggest a novel approach that mines image collections for shortest paths. While previous graph-based approaches [6] were focused on estimating the *importance* of target images based on their connection strength (similar to Google's *PageRank*), we exploit the image graph in a query-dependent fashion by discovering transitive similarities with the input image.

3. APPROACH

We first introduce some basic notation: a query image I_q is assumed to be given, for which pictures showing the same scene are to be retrieved from a dataset of other images $I_1, .., I_N$. We assume all pictures to belong to a finite set of *scenes* (indoor or outdoor locations), which is formalized by a latent *scene mapping* S from images I_i to their respective scene $S(i)$. Our goal is to retrieve images showing the same scene as I_q, i.e. a perfect retrieval score would be $\delta_{S(i),S(q)}$. Unfortunately, the scene mapping S is unknown. Instead, a similarity measure sim between images is given: for example, $sim(q, j)$ denotes the similarity between I_q and I_j. We assume a full matrix of similarities within the database to be given, i.e., $sim(i, j)$ is known for any choice of $(i, j) \in \{1, .., N\}^2$. For large databases, approximations can be found to work around this condition – see Section 5 for a discussion.

The similarity $sim(i, j)$ is derived from local correspondences (or *matches*) between I_i and I_j. Based on the number $m(i, j)$ of these matches, different similarity measures have been proposed in the literature (see [5] for a discussion). We use a simple normalization: for each image I_i, we compute the overall number of its correspondences with all other images: $n_i := \sum_j m(i, j)$. $sim(i, j)$ is obtained by normalizing $m(i, j)$ with respect to n_i and n_j:

$$sim(i, j) = \frac{N}{2} \cdot \left(\frac{m(i, j)}{n_i} + \frac{m(i, j)}{n_j} \right) \quad (1)$$

This measure puts the number of matches $m(i,j)$ in relation to the average number of matches both images share with all other pictures in the dataset. The factor N achieves a normalization with respect to the dataset size. Note also that the similarity measure is symmetric, i.e. $sim(i,j) = sim(j,i)$. Finally, as already indicated, we will employ clustering as one strategy to improve retrieval. The resulting clusters (which form a partitioning of the image indices) are denoted with $C_1, .., C_K \subset \{1, .., n\}$.

3.1 Features and Matching

Following the frequently used *patch-based* approach, we describe images as collections of local interest regions, which are matched to discover correspondences between pictures. For detecting local features, two different Interest Point (IP) detectors are utilized, namely the DoG detector [9] and the multi-scale Harris-Laplace detector [12]. Using these two complementary IP detectors, a good coverage of differently appearing image areas can be achieved. While the DoG detector is attracted by blob-like image content, the Harris-Laplace detector is strongly attracted by corners. The parameters of interest point detection were manually tuned to $thresh_{DoG} = 5.0$ and $harris_K = 0.08$. For the description of the local image content the SIFT descriptor [9] was used[1]. Matching the descriptors of two images was done using the SIFTGPU library[2]. The parameters *distmax=0.75* and *ratiomax=0.85* were optimized by a grid search maximizing the retrieval performance on a groundtruth dataset.

3.2 Approach 1: Cluster Matching

Our first approach to improve scene-based image retrieval exploits the structure of the image collection using clustering, a powerful tool which has already been applied to image datasets before [15, 20]. Our idea is that the resulting clusters coincide with scenes, such that by retrieving images from the right cluster we retrieve images from the right scene. We apply the following three-step procedure:

1. **clustering**: the image collection clustered into partitions $C_1, .., C_K$.
2. **cluster ranking**: given the query image I_q, we compute a similarity with each cluster, $sim^c(I_q, C_k)$, and order the clusters by descending similarity to I_q, obtaining a ranking $r_1, .., r_K$. The overall retrieval result is then obtained by a ranking on cluster level, i.e. all images from cluster C_{r_1} are ranked highest, then the ones from C_{r_2}, and so on.
3. **image ranking**: images within a retrieved cluster are ranked, too (which can have a strong influence on retrieval accuracy in case of large clusters).

Clustering.

We test different clustering methods for structuring image collections:

- *K-medoids*: As a simple baseline, we employ the well-known k-medoids clustering (using the implementation from the C clustering library[3]).
- *agglomerative clustering*: We test different versions of bottom-up agglomerative clustering, using the standard linkage methods "single", "average", and "complete". Again, the C clustering library was used.
- *spectral clustering*: We also test spectral clustering [10] using a projection with the top K eigenvectors of the normalized (asymmetric) graph laplacian, followed by K-means clustering.
- *ground truth*: Finally, we also test a system where the clusters coincide with the true partitioning of the dataset, i.e. for each scene there is a cluster containing all images from this scene (and only them). Though this information is likely not available in practice, we will use this setup as a control run indicating an upper bound for the performance of cluster matching.

A key problem with clustering is to determine the number of clusters K. To choose it, we test multiple choices of K and adopt the one maximizing Pearson's measure [14] (which has been used for image clustering before by Philbin et al. [15]): denoting with $A(C_1, C_2) := \sum_{i \in C_1} \sum_{j \in C_2} sim(i,j)$ the sum of similarities between two image sets, and with $C = \{1, .., N\}$ the whole image set, the quality of a cluster structure is defined as:

$$Q(C_1, .., C_k) = \sum_{k=1}^{K} \left(\frac{A(C_k, C_k)}{A(C,C)} - \frac{A(C_k, C)^2}{A(C,C)^2} \right) \quad (2)$$

Cluster Ranking.

For cluster ranking, we define an image-cluster similarity measure $sim^c(I_q, C_k)$, which is based on the number of correspondences between the query image I_q and images within C_k. We test two variants:

- **voting**: the similarities between I_q and all images in the cluster are accumulated (normalized by the cluster size). Let sim denote the image similarity measure from Equation (1):

$$sim^c(I_q, C_k) := \frac{1}{|C_k|} \sum_{I_i \in C_k} sim(I_q, I_i)$$

- **closest**: this strategy adopts the similarity to the *closest* image in the cluster as the cluster similarity

$$sim^c(I_q, C_k) := \min_{I_i \in C_k} sim(I_q, I_i)$$

Image Ranking.

Images I_i within a cluster can be ranked by employing the similarity measure $sim(I_q, I_i)$ (Equation (1)). An alternative will be introduced in the following section, where pictures are scored based on the minimal cost of reaching them via a *path* through the image collection.

3.3 Approach 2: Shortest Path Retrieval

A key problem with scene-based retrieval is that target images may share no direct correspondences with the query. However, there may be *indirect* correspondences, as illustrated in Figure 1 (where image B shares strong correspondences with A, which again is connected to the query). In the following, we introduce a probabilistic model that makes

[1] using the vlfeat library: http://www.vlfeat.org
[2] http://www.cs.unc.edu/~ccwu/siftgpu/
[3] bonsai.hgc.jp/~mdehoon/software/cluster/software.htm

use of these indirect matches. We model the retrieval score of a target image I_i by a probability that I_i shows the same scene as I_q. To overcome the lack of direct correspondences between I_i and I_q, this probability should be high if there is a *path* of images I_q and I_i in which all pairs of neighbor images are likely to show the same scene.

Probabilistic Image Similarity.

To realize shortest path search, we first model the probability that two subsequent images I_i, I_j in a path do show the same scene. We base this probability on the correspondence strength $sim(i,j)$, i.e. we compute $P[S(i)=S(j) \mid sim(i,j)]$. An estimate of this probability is illustrated in Figure 2: the higher the image similarity $sim(i,j)$ (x-axis), the higher we expect the target probability to be. The green solid curve was obtained by dividing the range of image similarities into bins and counting for each bin the percentage of associated image pairs from a test dataset truly showing the same scene. We see that our probability curve approximates a sigmoidal shape (the dashed line shows a fitted sigmoid).

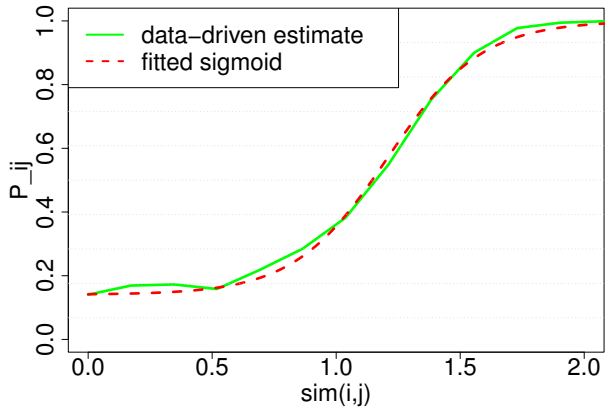


Figure 2: An estimate of the probability P_{ij} that two images I_i, I_j show the same scene, given their similarity $sim(i,j)$. The solid line shows an estimate on the "Porn 1" dataset (see Section 4), the dashed line a fitted sigmoid.

This observation can also be motivated theoretically: let us assume that if two images do *not* show the same scene, the number of their correspondences (and with it the similarity sim) is normally distributed, i.e. $p[sim(i,j) \mid S(i) \neq S(j)] = \mathcal{N}(\mu_0, \sigma_0)$. On the other hand, if both pictures *do* show the same scene, they may share some visual overlap or not (e.g., if they show different parts). Therefore, we use a Gaussian mixture: $p[sim(i,j) \mid S(i) = S(j)] = P_0 \cdot \mathcal{N}(\mu_0, \sigma_0) + P_1 \cdot \mathcal{N}(\mu_1, \sigma_1)$. By applying Bayes' rule, it can be shown that $P[S(i)=S(j) \mid sim(i,j)]$ follows a sigmoidal shape:

$$\begin{aligned} P_{ij} &:= P[S(i)=S(j) \mid sim(i,j)] \\ &= (1-\alpha) + \frac{\alpha}{1 + exp\{\beta + \gamma \cdot sim(i,j)\}} \end{aligned} \quad (3)$$

In the following, we will abbreviate this probability with P_{ij}, and will assume that an estimate of its parameters α, β, γ is given. We derive this estimate from the target dataset as illustrated in Figure 2, i.e. a sigmoid is fitted to a line based on estimates from all image pairs. Note that this requires the scene distribution $S(i)$ to be known – as this is unlikely to be the case in a practical setting, Section 4 will also show that learning the sigmoid generalizes well, i.e. we can fit it on one dataset and apply the learned parameters to another image collection with only minor performance loss.

Shortest Path Modeling.

Given an estimate of the probability that two images in a path show the same scene, we can formulate probabilistic retrieval scores based on shortest paths. We motivate this by marginalizing over all possible paths over images starting at I_q and ending at I_i. A path $\mathbf{p} = (i_1, i_2.., i_{L(\mathbf{p})})$ is defined as a sequence of indexes associated with images. $L(\mathbf{p})$ denotes the length of this sequence, and the start and end indices $i_1, i_{L(\mathbf{p})}$ equal q and i. We now model the probability that two images connected by a path $\mathbf{p}$ show the same scene, using the sigmoidal probability estimate from Equation (3):

$$P[S(i)=S(j), \mathbf{p}] \propto \prod_{l=1}^{L(\mathbf{p})-1} P_{i_l i_{l+1}}$$

By denoting the set of all possible paths from I_q to I_i with $\mathcal{P}(q,i)$, we can marginalize over all possible paths and compute the final retrieval score:

$$\begin{aligned} sim^{\mathcal{P}}(I_q, I_i) &:= P[S(q)=S(i)] \\ &= \sum_{\mathbf{p} \in \mathcal{P}(q,i)} P[S(q)=S(i), \mathbf{p}] \\ &\propto \sum_{\mathbf{p} \in \mathcal{P}(q,i)} \prod_{l=1}^{L(\mathbf{p})-1} P_{i_l i_{l+1}} \\ &\approx \max_{\mathbf{p} \in \mathcal{P}(q,i)} \prod_{l=1}^{L(\mathbf{p})-1} P_{i_l i_{l+1}} \\ &= \min_{\mathbf{p} \in \mathcal{P}(q,i)} \sum_{l=1}^{L(\mathbf{p})-1} -\log P_{i_l i_{l+1}} \end{aligned} \quad (4)$$

i.e. we define a graph with images as nodes and edges w_{ij} associated with costs (here, $-\log P_{ij}$). Given this graph, we compute the score $sim^{\mathcal{P}}(q,i)$ for a database image I_i by detecting a minimum cost path from I_q to I_i. To find these paths, an A^* search is used, which may (for efficiency reasons) be restricted to paths of a limited length.

3.4 Combining the two Approaches

We can apply shortest path retrieval globally on the whole collection $I_1, .., I_N$ by ranking images with the shortest path similarity measure $sim^{\mathcal{P}}$ (Equation (4)). Alternatively, we can combine it with cluster matching (Section 3.2) by replacing the standard similarity sim with $sim^{\mathcal{P}}$ in the image ranking step. The resulting procedure matches the query image first with all clusters, and then ranks pictures inside each cluster by a shortest path search. This comes with scalability improvements, as finding shortest paths is cheaper within small clusters as compared to the whole collection. We will evaluate this combined approach in Section 4.

4. EXPERIMENTS

In the following, we evaluate the two proposed approaches – cluster matching and shortest path retrieval – in quantitative experiments. We will first describe the experimental

protocol (Section 4.1). After this, we evaluate cluster matching (Section 4.2), followed by experiments with shortest path retrieval and a combination of both (Section 4.3). Finally, we test how well our approach – which requires some extra supervision compared to plain similarity matching – generalizes to entirely new datasets (Section 4.4).

4.1 Setup

We test our approach on several image datasets showing a mix of indoor and outdoor locations. All benchmarks consist of image series taken at the same indoor or outdoor location, whereas the level of occlusion may vary strongly, as well as scene texture and variations in perspective and illumination.

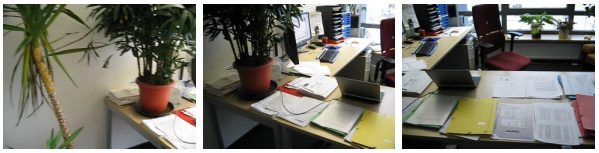
- **Offices**: We captured the first dataset ourselves by panning and zooming with a video camera through several offices in our lab. The dataset consists of 12 short clips corresponding to 12 offices, from which 576 keyframes were extracted by a regular sampling over time. The number of keyframes per series ranges from 38 to 59.

- **Rome**: This dataset of outdoor locations contains 1,400 pictures downloaded from *panoramio.com*. The dataset contains 7 series corresponding to 7 places in Rome (such as *Piazza del Popolo* or *Piazza di Spagna*). Each series contains 200 images showing buildings from the corresponding place. Illumination may vary strongly, as well as occlusion and perspective.

As another target domain, we choose the forensic investigation of crime scenes: here, an issue of particular interest to investigators is to uncover links between different images taken at the same crime scene. This allows to relate different criminal investigations (e.g., cases of child sexual abuse).

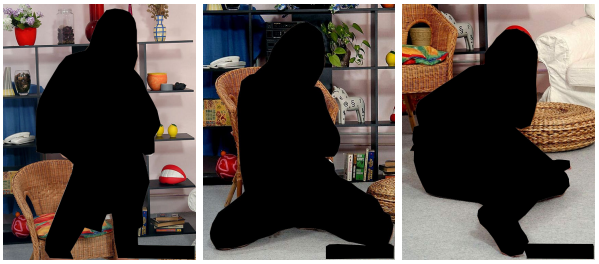
- **Porn 1+2**: These two datasets were collected from the web by different application partners in the forensic area. They contain series of pornographic pictures, whereas images from a series correspond to a *shooting*, showing the same indoor scene from varying perspectives and with varying occlusion. Pictures in a series show the same persons, but pose and scale may vary strongly. Logos (e.g., website names) were removed to avoid a biased evaluation. The first (less challenging) dataset **Porn 1** consists of 562 pictures in 37 series ranging from size 10 to size 18. Images show moderate variation in pose and perspective, and contain many highly textured scenes. The second (more challenging) dataset **Porn 2** contains 2,000 pictures in 200 series of size 10 [1]. As Figure 3(d) illustrates, this dataset contains significant occlusion by foreground objects, less textured backgrounds, and strong variation in pose and perspective.

For efficiency reasons, we conducted our evaluation in a leave-one-out fashion: for each dataset, the complete set of images was used for clustering and for estimation of the parameters of shortest path search α, β, γ (Equation (3)). Then, different settings were tested by performing retrieval for each image *after* removing it from the collection. We measure retrieval accuracy by the precision at a rank adapted to the series size (e.g., PREC@10 for Porn 2), averaged over all images in the respective dataset.

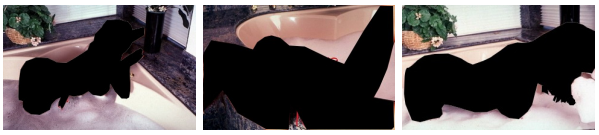


(a) Office

(b) Rome



(c) Porn 1



(d) Porn 2

Figure 3: Sample pictures from our datasets, showing mixed content of indoor and outdoor scenes with varying occlusion and variation of perspective and illumination (pornographic foregrounds were removed for illustration only).

4.2 Cluster Matching

We first evaluate cluster matching (Section 3.2). As a baseline, a standard approach based on direct correspondences is used, which ranks result images I_i by their similarity to the query image $sim(q, i)$ (Equation (1)). We compare this baseline with the extension described in Section 3.2, where clustering is applied and images are retrieved in order of their clusters.

Figure 4 illustrates quantitative results on all datasets. The dashed gray lines indicate the performance of our baseline system (bottom). The red curves correspond to the performance of cluster matching plotted over different values of the number of clusters K. Average linkage clustering was used, which was the most successful clustering technique. We see that for very few large clusters, the performance of cluster matching equals the baseline (which is not surprising, as images within clusters are ranked by the same similarity measure). The same holds true if the number of clusters increases towards the number of images, where cluster ranking is again based on the same similarity measure. For intermediate numbers of clusters, however, we observe that cluster matching gives performance improvements, e.g. on Porn 1 accuracy increases from 69.6% (baseline) to 84.2%.

Table 1: Comparing different clustering techniques for cluster matching. Only average linkage clustering gives stable improvements over the baseline.

	Offices - PREC@50 -		Rome - PREC@200 -		Porn 1 - PREC@10 -		Porn 2 - PREC@10 -	
clustering	K true	K est.	K true	K est.	K true	K est.	K true	K est.
baseline	0.315		0.244		0.696		0.300	
ground truth	0.882	—	0.812	—	0.902	—	0.624	—
k-medoids	0.189	0.404	0.172	0.264	0.418	0.436	0.177	0.172
single linkage	0.377	0.373	0.244	**0.319**	0.697	0.811	0.298	0.318
average linkage	**0.414**	**0.415**	0.244	0.287	**0.806**	**0.818**	**0.343**	**0.332**
complete linkage	0.278	0.358	0.219	0.246	0.693	0.693	0.292	0.274
spectral clustering	0.352	0.352	**0.263**	0.273	0.785	0.639	0.285	0.213

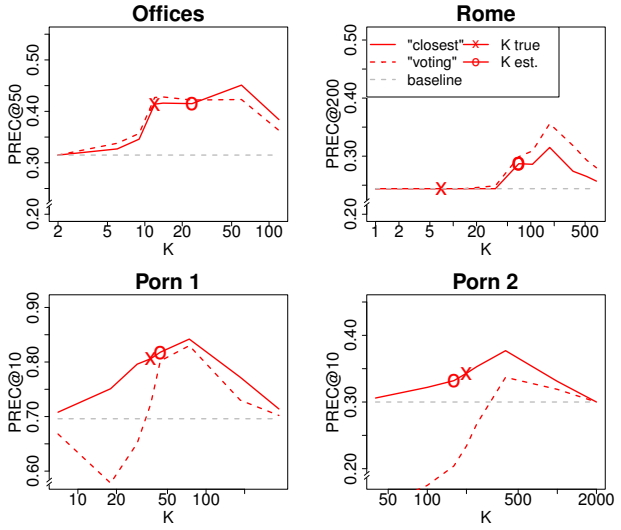


Figure 4: Cluster matching with average linkage gives performance improvements over a standard baseline, particularly if using the "closest" cluster ranking.

Comparing both cluster ranking strategies, we see that the "closest" matching strategy works better in most cases (with the "Rome" dataset being the only exception). Obviously, we can often find a single very similar image for matching to the right scene, while the "voting" strategy is influenced by matching with other images from the cluster showing different scenes or different parts of the right scene. We observed this across all clustering techniques, which is why results in the following will be presented for "closest" cluster ranking only.

Figure 4 also indicates the true number of clusters and an automatic estimate using Pearson's index (Equation (2)). We see that the estimate is reasonably close to the true number of clusters, and that stable improvements can be achieved with it. However, we also observe that in all cases performance could be improved further by a stronger oversegmentation of the dataset into even more clusters. This may be due to inaccuracies of clustering, as matching large clusters tends to retrieve significant amounts of mixed content while small, low-entropy clusters show higher purity.

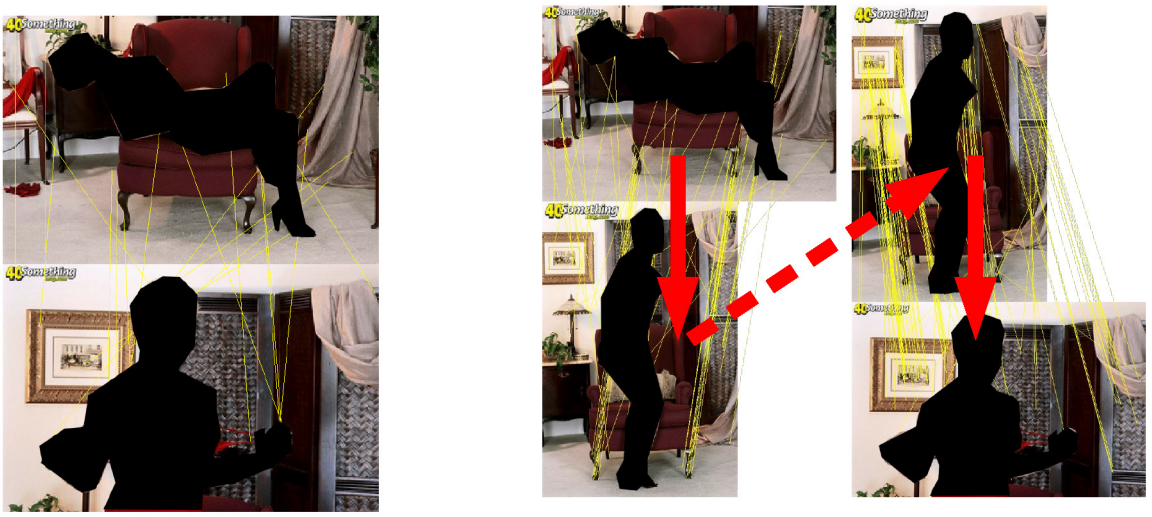
Table 1 illustrates results when using different clustering techniques (the "closest" matching strategy was used). We see that performance depends strongly on the clustering approach: K-medoids and complete linkage fail (as images from the same scene may share no common objects, we cannot expect compact clusters). Spectral clustering and single linkage give improvements in some cases, but only average linkage clustering gives stable improvements over the baseline on all datasets.

4.3 Shortest Path Retrieval

In the next experiment, we evaluate shortest path retrieval (Section 3.3), i.e. we do not only retrieve images most similar to the query image, but images which are connected to the query image via a low-cost path through the image collection. We start with a retrieval example in Figure 5, which illustrates correspondences between a query and a target image showing the same scene. Direct retrieval fails, as the two pictures share only a few false positive matches. On the other hand, shortest path search discovers a path via an intermediate image with strong correspondences to both query and target. The resulting path comes with low costs, such that the target image can be retrieved. As already outlined in Section 3.4, we can either apply shortest path retrieval by itself (i.e., on the whole collection), or we can combine it with cluster matching by employing it for ranking images within a cluster. Figure 6 gives quantitative retrieval results for either of these setups (average linkage clustering was used with K set to the estimate in Figure 4, and the maximum path length set to 5). We observe that shortest path retrieval leads to improvements, both when ranking images globally and within clusters. These improvements range from 3.2% ("Rome", with clustering) to 40.6% ("Office", no clustering). No clear statement can be made whether combining both approaches is beneficial – on "Porn 1+2", clustering combined with shortest paths gives the best results, while on "Offices" and "Rome" it is beneficial to apply shortest paths on the whole collection.

4.4 Generalization to Different Datasets

Compared to a plain direct matching, shortest path retrieval requires some extra supervision, as the parameters of the sigmoidal P_{ij} (Equation (3)) are learned from image series. In this section, we investigate how robust the learned parameters are with respect to a change of the dataset: shortest path retrieval is applied by *not* learning the parameters of P_{ij} on the respective dataset but on another one.



(a) (b)

Figure 5: A retrieval example on the "Porn 1" dataset (foreground removed from all images): (a) the target image (bottom) is not retrieved by the baseline, because it shares only a few (false positive) correspondences with the query image (top). (b) Shortest path retrieval succeeds by finding an intermediate image sharing strong correspondences with both the query (top left) and the target image (bottom right).

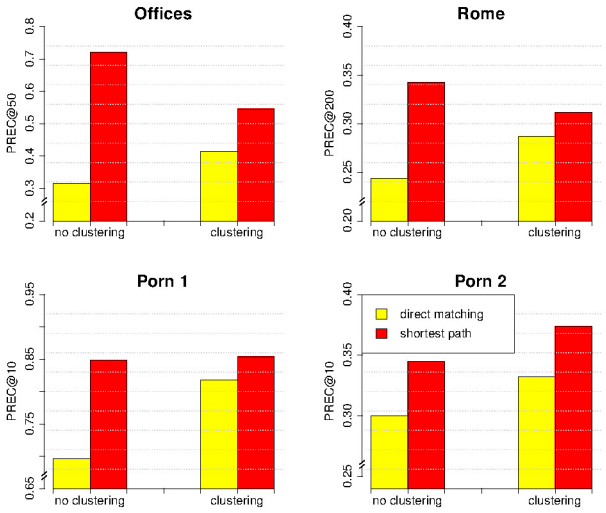


Figure 6: Shortest path retrieval leads to significant improvements in accuracy, both when applied on the whole collection and when combined with cluster matching.

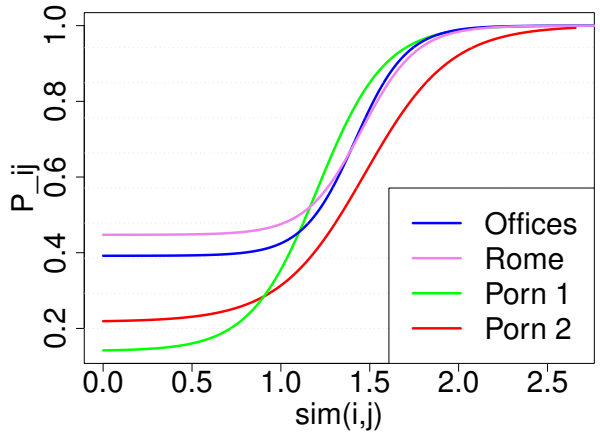
An illustration of the learned sigmoids is given in Figure 7: despite the diversity of the datasets, the sigmoids show a similar shape (differences for low similarities can be attributed to different numbers of series in each dataset, leading to different priors). Retrieval results are also illustrated in Figure 7. We tested shortest path retrieval both for *average linkage* clustering (with K set to the true number of series) and when using no clustering (i.e., shortest paths were found on the whole collection). Our key observation is that retrieval accuracy remains stable when training the system on another dataset (results when training on the three other datasets were averaged). Retrieval tends to be more accurate when training on the correct dataset, but in most cases only minor performance drops occur (0.5% on average). This indicates that shortest path retrieval generalizes well to different datasets and is applicable in practice.

5. CONCLUSIONS

To improve the accuracy of scene-based image retrieval, we have proposed two strategies of dealing with target images sharing no direct correspondences with the query, namely cluster matching and shortest path retrieval. By exploiting the structure of the targeted image collection, particularly shortest path retrieval was demonstrated to give improvements over standard direct image matching on different datasets of indoor and outdoor locations.

In the future, we plan to overcome scalability issues to improve the practical applicability of our approach: for large-scale datasets, a full similarity matrix (as we assume to be given) may not be available, and the search for shortest paths may become inefficient. Here, a key strategy lies in the combination of cluster matching with shortest paths. This was already demonstrated to give a good accuracy in Section 4. Also, by applying shortest path search *within* clusters, it can be conducted much more efficiently (clustering has already been demonstrated to be applicable at a large scale [15]). Solutions to overcome the need for full similarity matrices may include the use index structures for patch matching, like vector quantization in combination with inverted files [18].

Another interesting direction might be to make stronger use of spatial position. While the presented work is entirely based on patch appearance, multiple extensions for direct image matching have been proposed that take the



no clustering				
training / testing	Offices	Rome	Porn 1	Porn 2
same dataset	0.721	0.343	0.849	0.345
other dataset (⌀)	0.721	0.341	0.848	0.344
avg. linkage clustering				
training / testing	Offices	Rome	Porn 1	Porn 2
same dataset	0.617	0.343	0.861	0.375
other dataset (⌀)	0.584	0.342	0.867	0.370

Figure 7: Top: Sigmoids learned on different datasets for modeling P_{ij}. Bottom: When training the system on a different dataset than testing it on, performance remains stable.

consistency of feature positions into account, ranging from global geometric transformations [9] to local patch constellations [18]. These approaches might be extended to transitive matching, requiring feature positions to be consistent over the complete path of images from query to target.

Acknowledgement

This research was funded by BMBF grant INBEKI 13*N*10787.

6. REFERENCES

[1] M. Ali and J. Hofmann. An Extensive Approach to Content Based Image Retrieval Using Low- & High-Level Descriptors. Technical report, Master's Thesis, IT University of Gothenburg, Sweden, 2006.

[2] J. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building Rome on a Cloudless Day. In *Proc. ECCV*, pages 368–381. 2010.

[3] L. V. Gool, T. Tuytelaars, and A. Turina. Local Features for Image Retrieval. In *State-of-the-Art in Content-Based Image and Video Retrieval*, pages 21–41. Kluwer Academic Publishers, 2001.

[4] A. Jain, J.-E. Lee, R. Jin, and N. Gregg. Content-based Image Retrieval: An Application to Tattoo Images. In *Proc. ICIP*, pages 2745 –2748, 2009.

[5] H. Jegou, M. Douze, and C. Schmid. Hamming Embedding and Weak Geometry Consistency for Large Scale Image Search - Extended Version. Technical report, INRIA, RR 6709, 2008.

[6] Y. Jing and S. Baluja. VisualRank: Applying PageRank to Large-Scale Image Search. *IEEE Trans. PAMI*, 30(11):1877–1890, 2008.

[7] Z. Li-jia, Z. Shao-min, Z. Da-zhe, Z. Hong, and L. Shu-kuan. Medical Image Retrieval Using SIFT Feature. In *Proc. CISP*, pages 1 –4, 2009.

[8] D. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proc. ICCV*, pages 1150–1157, 1999.

[9] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004.

[10] U. Luxburg. A Tutorial on Spectral Clustering. *Statistics and Computing*, 17:395–416, 2007.

[11] K. Mikolajczyk. A Performance Evaluation of Local Descriptors. In *Proc. CVPR*, pages 257–263, 2003.

[12] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, 2004.

[13] S. Nayar and H. M. ad S. Nene. *Parametric Appearance Representation*, pages 131–160. Oxford Univ. Press, 1996.

[14] M. Newman and M. Girvan. Finding and Evaluating Community Structure in Networks. *Physical Review*, 69(2), 2004.

[15] J. Philbin and A. Zisserman. Object Mining Using a Matching Graph on Very Large Image Collections. In *Proc. Indian Conf. on Comp. Vis., Graphics & Img. Proc.*, pages 738–745, 2008.

[16] Microsoft Photosynth. available from `http://photosynth.net/` (retrieved: December 2010).

[17] T. Quack, B. Leibe, and L. van Gool. World-scale Mining of Objects and Events from Community Photo Collections. In *Proc. CIVR*, pages 47–56, 2008.

[18] J. Sivic and A. Zisserman. Video Google: Efficient Visual Search of Videos. In *Toward Category-Level Object Recognition*, pages 127–144. Springer-Verlag New York, Inc., 2006.

[19] D. Smith and R. Harvey. Document retrieval using image features. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 47–51, New York, NY, USA, 2010. ACM.

[20] T. Tuytelaars, C. Lampert, M. Blaschko, and W. Buntine. Unsupervised Object Discovery: A Comparison. *Int. J. Comp. Vis.*, 88(2):284–302, 2010.

[21] X. Wangming, W. Jin, L. Xinhai, Z. Lei, and S. Gang. Application of Image SIFT Features to the Context of CBIR. In *Proc. CSSE*, pages 552 –555, 2008.