# Automatic Query Image Disambiguation for Content-based Image Retrieval

Björn Barz and Joachim Denzler

*Computer Vision Group, Friedrich Schiller University Jena, Ernst-Abbe-Platz 2, 07743 Jena, Germany*

Keywords:     Content-based Image Retrieval, Interactive Image Retrieval, Relevance Feedback.

Abstract:     Query images presented to content-based image retrieval systems often have various different interpretations, making it difficult to identify the search objective pursued by the user. We propose a technique for overcoming this ambiguity, while keeping the amount of required user interaction at a minimum. To achieve this, the neighborhood of the query image is divided into coherent clusters from which the user may choose the relevant ones. A novel feedback integration technique is then employed to re-rank the entire database with regard to both the user feedback and the original query. We evaluate our approach on the publicly available MIRFLICKR-25K dataset, where it leads to a relative improvement of average precision by 23% over the baseline retrieval, which does not distinguish between different image senses.

## 1 INTRODUCTION

Content-Based Image Retrieval (CBIR) refers to the task of retrieving a ranked list of images from a potentially large database that are semantically similar to one or multiple given query images. It has been a popular field of research since 1993 (Niblack et al., 1993) and its advantages over traditional image retrieval based on textual queries are manifold: CBIR allows for a more direct and more fine-grained encoding of what is being searched for using example images and avoids the cost of textual annotation of all images in the database. Even in cases where such describing texts are naturally given (*e.g.*, when searching for images on the web), the description may lack some aspects of the image that the annotator did not care about, but the user searching for that image does.

In some applications, specifying a textual query for images may even be impossible. An example is biodiversity research, where the class of the object on the query image is unknown and to be determined using similar images retrieved from an annotated database (Freytag et al., 2015). Another example is flood risk assessment based on social media images (Poser and Dransch, 2010), where the user searches for images that allow for estimation of the severity of a flood and the expected damage. This search objective is too complex for being expressed in the form



Figure 1: An example of query image ambiguity. Given the query image in the center, the user might be searching for any of the topics listed around it, which do all appear intermixed in the baseline retrieval results. All images are from the MIRFLICKR-25K dataset (Huiskes and Lew, 2008).

of keywords (*e.g.*, "images showing street scenes with cars and traffic-signs partially occluded by polluted water") and, hence, has to rely on query-by-example approaches.

In the recent past, there has been a notable amount of active research on the special case of *object* or *instance retrieval* (Jégou et al., 2010; Arandjelović and Zisserman, 2012; Jégou and Zisserman, 2014; Babenko and Lempitsky, 2015; Yu et al., 2017; Gordo et al., 2016), which refers to the task of only retrieving images showing exactly the same object as the query image. Approaches for solving this problem

249

have reached a mature performance on the standard object retrieval benchmarks recently thanks to end-to-end learned deep representations (Gordo et al., 2016).

However, in the majority of search scenarios, users are not looking for images of exactly the same object, but for images similar, but not identical to the given one. This involves some ambiguity inherent in the query on several levels (an example is given in Figure 1):

1. Different users may refer to different regions in the image. This problem is evident if the image contains multiple objects, but the user may also be looking for a specific part of a single object.

2. If the user is searching for images showing objects of the same class as the object in the query image, the granularity of the classification in the user's mind is not known to the system. If the query showed, for example, a poodle, the user may search for other poodles, dogs, or animals in general.

3. A single object may even belong to multiple orthogonal classes. Given a query image showing, for instance, a white poodle puppy, it is unclear whether the user is searching for poodles, for puppies, for white animals, or for combinations of those categories.

4. The visual aspect of the image that constitutes the search is not always obvious. Consider, for example, an oil painting of a city skyline at night as query image. The user may search for other images of cities, but she might also be interested in images taken at night or in oil paintings regardless of the actual content.

Given all these kinds of ambiguity, it is often impossible for an image retrieval system to provide an accurate, satisfactory answer to a query consisting of a single image without any further information. Many CBIR systems hence enable the user to mark relevant and sometimes also irrelevant images among the initially retrieved results. This relevance feedback is then used to issue a refined query (Rocchio, 1971; Jin and French, 2003; Deselaers et al., 2008). This process, however, relies on the cooperation and the patience of the user, who may not be willing to go through a large set of mainly irrelevant results in order to provide extensive relevance annotations.

In this work, we present an approach to simplify this feedback process and reduce the user's effort to a minimum, while still being able to improve the relevance of the retrieved images significantly. Our method consists of two steps: First, we automatically identify different meanings of the query image through clustering of the highest scoring retrieval results. The user may select one or more relevant clusters based on a few preview images shown for each cluster. We then apply a novel re-ranking technique that adjusts the scores of all images in the database with respect to this simple user feedback. Note that the number of clusters to choose from will be much smaller than the number of images the user would have to annotate for image-wise relevance feedback.

Our re-ranking technique adjusts the effective distance of database images from the query, so that images in the same direction from the query as the selected cluster(s) are moved closer to the query and images in the opposite direction are shifted away. This avoids error-prone hard decisions for images from a single cluster and takes both the similarity to the selected cluster and the similarity to the query image into account.

For all hyper-parameters of the algorithm, we propose either appropriate default values or suitable heuristics for determining them in an unsupervised manner, so that our method can be used without any need for hyper-parameter tuning in practice.

The remainder of this paper is organized as follows: We briefly review related work on relevance feedback in image retrieval and similar clustering approaches in Section 2. The details of our proposed *Automatic Query Image Disambiguation (AID)* method are set out in Section 3. Experiments described in Section 4 and conducted on a publicly available dataset of 25,000 Flickr images (Huiskes and Lew, 2008) demonstrate the usefulness of our method and its advantages over previous approaches. Section 5 summarizes the results.

## 2 RELATED WORK

The incorporation of relevance feedback has been a popular method for refinement of search results in information retrieval for a long time. Typical approaches can be divided into a handful of classes: **Query-Point Movement (QPM)** approaches adjust the initial query feature vector by moving it towards the direction of selected relevant images and away from irrelevant ones (Rocchio, 1971). Doing so, however, they assume that all relevant images are located in a convex cluster in the feature space, which is rarely true (Jin and French, 2003). On the other hand, approaches based on **distance or similarity learning** optimize the distance metric used to compare images, so that the images marked as relevant have a low pairwise distance, while having a rather large distance to the images marked as irrelevant (Ishikawa et al., 1998; Deselaers et al., 2008). In the simplest case, the metric learning may consist in just re-weighting

the individual features (Deselaers et al., 2008). Speaking of machine learning approaches, **classification** techniques are also often employed to distinguish between relevant and irrelevant images in a binary classification setting (Guo et al., 2002; Tong and Chang, 2001). Finally, **probabilistic** approaches estimate the distribution of a random variable indicating whether a certain image is relevant or not, conditioned by the user feedback (Cox et al., 2000; Arevalillo-Herráez et al., 2010; Glowacka et al., 2016).

However, all those approaches require the user to give relevance feedback regarding several images, which often has to be done repeatedly for successive refinement of retrieval results. Some methods even need more complex feedback than binary relevance annotations, asking the user to assign a relevance *score* to each image (Kim and Chung, 2003) or to annotate particularly important regions in the images (Freytag et al., 2015).

In contrast, our approach keeps the effort on the user's side as low as possible by restricting feedback to the selection of a single cluster of images. Resolving the ambiguity of the query by clustering its neighborhood has been successfully employed before, but very often relies on textual information (Zha et al., 2009; Loeff et al., 2006; Cai et al., 2004), which is not always available. One exception is the CLUE method (Chen et al., 2005), which relies solely on image features and is most similar to our approach. In opposition to CLUE, which uses spectral clustering for being able to deal with non-metric similarity measures, we rely on k-Means clustering in Euclidean feature spaces, so that we can use the centroids of the selected clusters to refine the retrieval results.

A major insufficiency of CLUE and other existing works is that they fail to provide a technique for incorporating user feedback regarding the set of clusters provided by the methods. Instead, the user has to browse all clusters individually, which is not optimal for several reasons: First, similar images near the cluster boundaries are likely to be mistakenly located in different clusters and, second, a too large number of clusters will result in the relevant images being split up across multiple clusters. Moreover, the overall set of results is always restricted to the initially retrieved neighborhood the query.

Our approach is, in contrast, able to re-rank the entire dataset with regard to the selection of one or more clusters in a way that avoids hard decisions and takes both the distance to the initial query and the similarity to the selected cluster into account.

# 3 AUTOMATIC QUERY IMAGE DISAMBIGUATION (AID)

Our automatic query image disambiguation method (AID) consists of two parts: The unsupervised identification of different meanings inherent in the query image (cf. Section 3.1), from which the user may then choose relevant ones, and the refinement of the retrieval results according to this feedback (cf. Section 3.2). The entire process is illustrated exemplarily in Figure 2.

## 3.1 Identification of Image Senses

In the following, we assume all images to be represented by $d$ real-valued features, which could be, for example, *neural codes* extracted from a neural network (cf. Section 4.1). Given a query image $q \in \mathbb{R}^d$ and a database $\mathcal{B} \subset \mathbb{R}^d$ with $n := |B|$ images, we first retrieve the $m$ nearest neighbors $X = \{x_1, x_2, \ldots, x_m\} \subseteq \mathcal{B}$ of $q$ from the database. We employ the Euclidean distance for this purpose, which has been shown to be a reasonable dissimilarity measure when used in combination with semantically meaningful feature spaces (Babenko et al., 2014; Yu et al., 2017; Gordo et al., 2016).

In the following, this step is referred to as *baseline retrieval* and will usually result in images that are all similar to the query, but with respect to different aspects of the query, so that they might not be similar compared to each other (cf. Figure 2a). We assume that database items resembling the same aspect of the query are located in the same direction from the query in the feature space. Thus, we first represent all retrieved neighbors by their direction from the query:

$$\hat{X} := \left\{ \underbrace{\frac{x_i - q}{\|x_i - q\|}}_{=: \hat{x}_i} \;\middle|\; i = 1, \ldots, m \right\}, \qquad (1)$$

where $\|\cdot\|$ denotes the Euclidean norm. Discarding the magnitude of feature vector differences and focusing on directions instead has proven to be beneficial for image retrieval, *e.g.*, as so-called triangulation embedding (Jégou and Zisserman, 2014).

We then divide these directions $\hat{X}$ of the neighborhood into $k$ disjoint clusters (cf. Figure 2b) using k-Means (Lloyd, 1982). For inspection by the user, each cluster is represented by a small set of $r$ images that belong to the cluster and are closest to the query $q$. This is in opposition to CLUE (Chen et al., 2005), which represents each cluster by its medoid. However, this makes it difficult for the user to assess the relevance of the cluster, since the medoid has no direct relation to the query anymore.
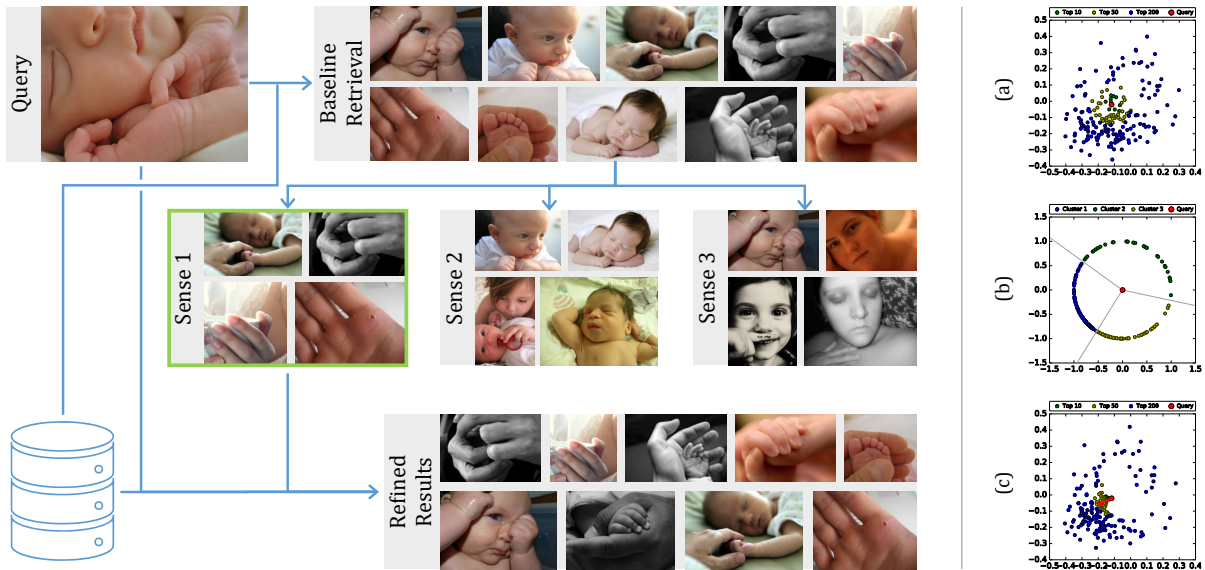
Figure 2: **Left:** Illustration of query image disambiguation and refinement of retrieval results. In this example, the user selected Sense 1 as relevant. **Right:** Schematic illustration of the data representation at each step in 2-d space.

The proper number $k$ of clusters depends on the ambiguity of the query and also on the granularity of the search objective, because, for instance, more clusters are needed to distinguish between poodles and cocker spaniels than between dogs and other animals. Thus, there is no single adequate number of clusters for a certain query, but the same fixed value for $k$ is also likely to be less appropriate for some queries than for others. We hence use a heuristic found in literature for determining a query-dependent number of clusters based on the largest *Eigengap* (Cai et al., 2004):

1. Construct an affinity matrix $A \in \mathbb{R}^{m \times m}$ with $A_{i,j} :=$ $\exp\left(-\eta \cdot \|\hat{x}_i - \hat{x}_j\|^2\right)$.

2. Compute the graph Laplacian $L = D - A$ with $D := \mathrm{diag}(s_1, s_2, \ldots, s_m)$, where $s_i := \sum_{j=1}^m A_{i,j}$.

3. Solve the generalized eigenvalue problem $L\mathbf{v} = \lambda D\mathbf{v}$ and sort the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ in ascending order, *i.e.*, $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$.

4. Set $k := \mathrm{argmax}_{1 \leq i < m}(\lambda_{i+1} - \lambda_i)$.

This heuristic has originally been used in combination with spectral clustering, where the mentioned eigenvalue problem has to be solved as part of the clustering algorithm. Here, we use it just for determining the number of clusters and then apply k-Means as usual. The hyper-parameter $\eta$ can be used to control the granularity of the clusters: a smaller $\eta$ will result in fewer clusters on average, while large $\eta$ will lead to more clusters. In our experiments we set $\eta = \sqrt{d}$ and cap the number of clusters at a maximum of 10 to limit the effort imposed on the user.

## 3.2 Refinement of Results

Given a selection of $\ell$ relevant clusters represented by their centroids[1] $C = \{c_1, \ldots, c_\ell\}$, we re-rank all images in the database by adjusting their effective distance to the query, so that images in the same direction as the selected clusters are moved closer to the query, while images in the opposite direction are shifted away and images in the orthogonal direction keep their original scores. The images are then sorted according to this adjusted distance (cf. Figure 2c).

Let $x \in \mathcal{B}$ denote any image in the database, $\delta(x) := \|x - q\|$ its Euclidean distance to the query (already computed during the initial retrieval) and $\sigma(x)$ the cosine similarity between the direction from $q$ to $x$ and from $q$ to the center of the relevant cluster closest to $x$, formally:

$$\sigma(x) := \max_{c_i \in C} \frac{c_i^\top (x - q)}{\|c_i\| \cdot \|x - q\|} . \qquad (2)$$

We define the adjusted distance score $\tilde{\delta}(x)$ of $x$ as

$$\tilde{\delta}(x) := \delta(x) - \mathrm{sign}(\sigma(x)) \cdot |\sigma(x)|^\gamma \cdot \beta, \qquad (3)$$

where $\beta > 0$ is a constant that we set to $\beta := \max_{x' \in \mathcal{B}} \delta(x')$ to ensure that even the most distant database item can be drawn to the query if it lies exactly in the selected direction. The hyper-parameter $\gamma \geq 0$ controls the influence of the user feedback: for $\gamma > 1$,

---

[1]Note that clustering has been performed on $\hat{X}$, so that the centroids represent (unnormalized) directions from the query as origin.

only the distances of images matching the selected direction more exactly will be adjusted, while for $\gamma < 1$ peripheral images are affected as well. We consider $\gamma = 1.0$ a good default and use this in our experiments.

Note that Equation (3) allows for "negative distances", but this is not a problem, because we use the adjusted distance just for ranking and it is not a proper pair-wise metric anyway due to its query-dependence.

# 4 EXPERIMENTS

## 4.1 Setup

**Dataset.** We evaluate the usefulness of our approach for image retrieval on the publicly available MIRFLICKR-25K dataset[2] (Huiskes and Lew, 2008), which consists of 25,000 images collected from Flickr. All images have been annotated with a subset of 24 predefined topics by human annotators, where a topic is assigned to an image if it is at least somewhat relevant to it ("wide sense annotations"). A second set of annotations links topics to images only if the respective topic is saliently present in the image ("narrow sense annotations"), but these annotations are only available for 14 topics. Note that a single image may belong to multiple topics, which is in accordance with the ambiguity of query images.

The median number of images assigned to such a "narrow sense" topic is 669, with the largest topic ("people") containing 7,849 and the smallest one ("baby") containing 116 images. Narrow sense topics are available for 12,681 images, which are on average assigned to 2 such topics, but at most to 5.

We use all of those images to define 25,002 test-cases: Each image is issued as individual query for each of its assigned topics and the implied goal of the imaginary user is to find images belonging to the same topic. Due to the inherent ambiguity of a single query image, relevance feedback will be necessary in most cases to accomplish this task.

**Image Representations.** Following the concept of *Neural Codes* (Babenko et al., 2014), we extract features for all images from a certain layer of a convolutional neural network. Specifically, we use the first fully-connected layer (`fc6`) of the VGG16 network (Simonyan and Zisserman, 2014) and reduce the descriptors to 512 dimensions using PCA. We do explicitly not use features from the convolutional layers, although they have been shown to be superior for *object* retrieval when aggregated properly (Babenko

---

[2]http://press.liacs.nl/mirflickr/

and Lempitsky, 2015; Zhi et al., 2016). This does, however, not hold for the quite different task of category retrieval, where the fully-connected layers—being closer to the class prediction layer and hence carrying more semantic information—provide better results (Yu et al., 2017).

**Evaluation Metrics.** Since the output of our image retrieval system is a ranked list of all images in the database, with the most relevant image at the top, we measure performance in terms of mean average precision (mAP) over all queries. Though this measure is adequate for capturing the quality of the entire ranking, it takes both precision and recall into account, whereas a typical user is seldom interested in retrieving all images belonging to a certain topic, but puts much more emphasis on the precision of the top results. Thus, we also report the precision of the top $\kappa$ results for $1 \leq \kappa \leq 100$.

Because k-Means clustering is highly initialization-dependent, we have repeated all experiments 5 times and report the mean value of each performance metric. The standard deviation of the results was less than 0.1% in all cases.

**Simulation of User Feedback.** We investigate two different scenarios regarding user feedback: In the first scenario, the user must select exactly one of the proposed clusters and we simulate this by choosing the cluster whose set of preview images has the highest precision. In the second scenario, the user may choose multiple or even zero relevant clusters, which we simulate by selecting all clusters whose precision among the preview images is at least 50%. If the user does not select any cluster, we do not perform any refinement, but return the baseline retrieval results.

A set of $r = 10$ preview images is shown for each cluster, since ten images should be enough for assessing the quality of a cluster and we want to keep the number of images the user has to review as low as possible. Note that our feedback simulation does not have access to all images in a cluster for assessing its relevance, but to those preview images only, just like the end-user.

**The Competition.** We do not only evaluate the gain in performance achieved by our AID method compared to the baseline retrieval, but also compare it with our own implementation[3] of CLUE (Chen et al., 2005), which uses a different clustering strategy. Since CLUE does not propose any method to

---

[3]The source code of our implementation of AID and CLUE is available at https://github.com/cvjena/aid.
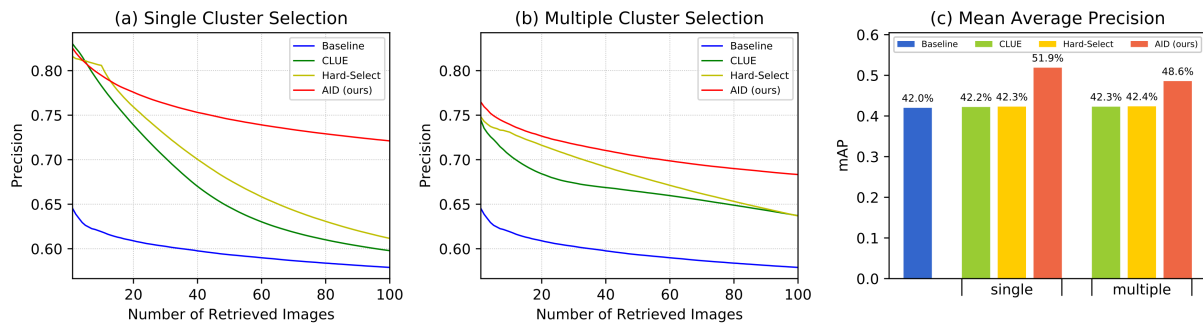
Figure 3: Performance of our AID approach compared with baseline retrieval, CLUE, and hard cluster selection on the same set of clusters as used by AID.

incorporate user feedback, we construct a refined ranking by simply moving the selected cluster(s) to the top of the list and then continuing with the clusters in the order determined by CLUE, which sorts clusters by their minimum distance to the query.

For evaluation of the individual contributions of both our novel re-ranking method on the one hand and the different clustering scheme on the other hand, we also evaluate hard cluster selection (as used by CLUE) on the same set of clusters as determined by AID. In this scenario, the selected clusters are simply moved to the top of the ranking, leaving the order of images within clusters unmodified.

The number $m$ of nearest neighbors of the query used as input in the clustering stage should be large enough to include images from all possible meanings of the query, but larger $m$ also imply higher computational cost. We choose $m = 200$ as a trade-off for both, CLUE and AID.

## 4.2 Quantitative Results

The charts in Figure 3 show that our AID approach is able to improve the retrieval results significantly, given a minimum amount of user feedback. Re-ranking the entire database is of great benefit compared with simply restricting the final retrieval results to the selected cluster. The latter is done by CLUE and precludes it from retrieving relevant images not contained in the small set of initial results. Therefore, CLUE can only keep up with AID regarding the precision of the top 10 results, but cannot improve the precision of the following results or the mAP significantly.

AID, in contrast, performs a global adjustment of the ranking, leading to a relative improvement of mAP over CLUE by 23% and of P@100 by 21%.

The results for hard cluster selection on the same set of clusters as used by AID reveal that applying k-Means on $\hat{X}$ instead of $X$ (directions instead of absolute positions) is superior to the clustering scheme used by CLUE. However, there is still a significant

gap of performance compared with AID, again underlining the importance of global re-ranking.

Interestingly, though AID can handle the selection of multiple relevant clusters, it cannot take advantage from it, but multiple clusters even slightly reduce its performance (cf. Figure 3b). This could not be remedied by varying γ either and could be attributed to the fact that AID considers all selected clusters to be equally relevant, which may not be the case. If only the most relevant cluster is selected, in contrast, other relevant clusters will benefit from the adjusted distances as well according to their similarity to the selected one. This is supported by the fact that AID using a single relevant cluster is still superior to all methods allowing the selection of multiple clusters. Thus, we can indeed keep the required amount of user interaction at a minimum—asking the user to select a single relevant cluster only—while still providing considerably improved results.

## 4.3 Qualitative Examples

For an exemplary demonstration of our approach, we applied AID with a fixed number of $k = 3$ clusters to the query image from Figure 1. The top 8 results from the refined ranking for each cluster are shown in Figure 4. It can easily be observed that all clusters capture different aspects of the query: The first one corresponds to the topic "hands", the second to "baby", and the third to "portrait".

Note that some images appear at the top of more than one refined ranking due to their high similarity to the query image. This is an advantage of AID compared with other approaches using hard cluster decisions, because the retrieved images might be just as ambiguous as queries and can belong to several topics. In this example, there is a natural overlap between the results for the topics "baby" and "portrait", but also between "baby" and "hands", since the hands of a baby are the prominent content of some images.

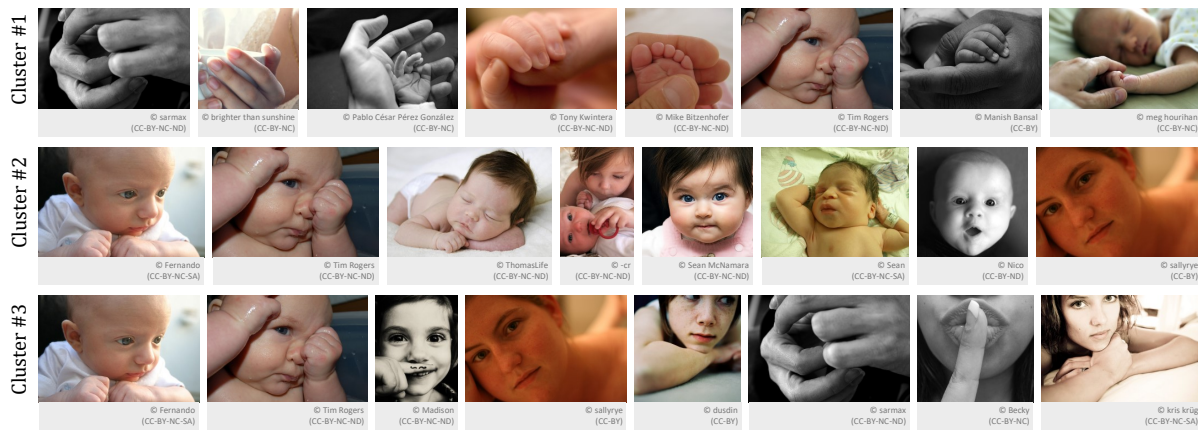A second example given in Figure 5 shows how

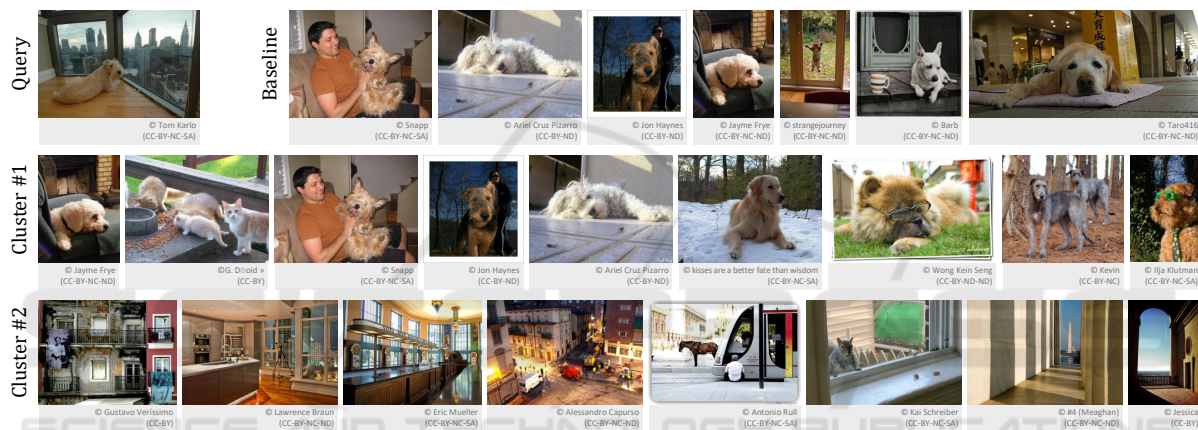Figure 4: Top refined results for the query from Figure 1.



Figure 5: Baseline and refined results for another query with two different meanings.

AID distinguishes between two meanings of another query showing a dog in front of a city skyline. While the baseline ranking focuses on dogs, the results can as well be refined towards city and indoor scenes.

## 5 CONCLUSIONS

We have proposed a method for refining content-based image retrieval results with regard to the users' actual search objective based on a minimal amount of user feedback. Thanks to automatic disambiguation of the query image through clustering, the effort imposed on the user is reduced to the selection of a single relevant cluster. Using a novel global re-ranking method that adjusts the distance of all images in the database according to that feedback, we considerably improve on existing approaches that limit the retrieval results to the selected cluster.

It remains as an open question, how feedback consisting of the selection of multiple clusters can be incorporated without falling behind the performance obtained from the selection of the single best cluster. Since some relevant clusters are more accurate than others, future work might investigate whether asking for a *ranking* of relevant clusters can be beneficial.

Furthermore, we are not entirely satisfied with the heuristic currently employed to determine the number of clusters, since it is inspired by spectral clustering, which we do not apply. Since query images are often, but not always ambiguous, it would also be beneficial to detect when disambiguation is likely to be not necessary at all.

# REFERENCES

Arandjelović, R. and Zisserman, A. (2012). Three things everyone should know to improve object retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2911–2918. IEEE.

Arevalillo-Herráez, M., Ferri, F. J., and Domingo, J. (2010). A naive relevance feedback model for content-based image retrieval using multiple similarity measures. *Pattern Recognition*, 43(3):619–629.

Babenko, A. and Lempitsky, V. (2015). Aggregating local deep features for image retrieval. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1269–1277.

Babenko, A., Slesarev, A., Chigorin, A., and Lempitsky, V. (2014). Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer.

Cai, D., He, X., Li, Z., Ma, W.-Y., and Wen, J.-R. (2004). Hierarchical clustering of www image search results using visual, textual and link information. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 952–959. ACM.

Chen, Y., Wang, J. Z., and Krovetz, R. (2005). Clue: cluster-based retrieval of images by unsupervised learning. *IEEE transactions on Image Processing*, 14(8):1187–1201.

Cox, I. J., Miller, M. L., Minka, T. P., Papathomas, T. V., and Yianilos, P. N. (2000). The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments. *IEEE transactions on image processing*, 9(1):20–37.

Deselaers, T., Paredes, R., Vidal, E., and Ney, H. (2008). Learning weighted distances for relevance feedback in image retrieval. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE.

Freytag, A., Schadt, A., and Denzler, J. (2015). Interactive image retrieval for biodiversity research. In *German Conference on Pattern Recognition*, pages 129–141. Springer.

Glowacka, D., Teh, Y. W., and Shawe-Taylor, J. (2016). Image retrieval with a bayesian model of relevance feedback. *arXiv:1603.09522*.

Gordo, A., Almazan, J., Revaud, J., and Larlus, D. (2016). End-to-end learning of deep visual representations for image retrieval. *arXiv:1610.07940*.

Guo, G.-D., Jain, A. K., Ma, W.-Y., and Zhang, H.-J. (2002). Learning similarity measure for natural image retrieval with relevance feedback. *IEEE Transactions on Neural Networks*, 13(4):811–820.

Huiskes, M. J. and Lew, M. S. (2008). The mir flickr retrieval evaluation. In *MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval*, New York, NY, USA. ACM.

Ishikawa, Y., Subramanya, R., and Faloutsos, C. (1998). Mindreader: Querying databases through multiple examples. *Computer Science Department*, page 551.

Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010). Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311. IEEE.

Jégou, H. and Zisserman, A. (2014). Triangulation embedding and democratic aggregation for image search. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3310–3317.

Jin, X. and French, J. C. (2003). Improving image retrieval effectiveness via multiple queries. In *ACM international workshop on Multimedia databases*, pages 86–93. ACM.

Kim, D.-H. and Chung, C.-W. (2003). Qcluster: relevance feedback using adaptive clustering for content-based image retrieval. In *ACM SIGMOD international conference on Management of data*, pages 599–610. ACM.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

Loeff, N., Alm, C. O., and Forsyth, D. A. (2006). Discriminating image senses by clustering with multimodal features. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 547–554. Association for Computational Linguistics.

Niblack, C. W., Barber, R., Equitz, W., Flickner, M. D., Glasman, E. H., Petkovic, D., Yanker, P., Faloutsos, C., and Taubin, G. (1993). Qbic project: querying images by content, using color, texture, and shape. In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, pages 173–187. International Society for Optics and Photonics.

Poser, K. and Dransch, D. (2010). Volunteered geographic information for disaster management with application to rapid flood damage estimation. *Geomatica*, 64(1):89–98.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*.

Tong, S. and Chang, E. (2001). Support vector machine active learning for image retrieval. In *ACM international conference on Multimedia*, pages 107–118. ACM.

Yu, W., Yang, K., Yao, H., Sun, X., and Xu, P. (2017). Exploiting the complementary strengths of multi-layer cnn features for image retrieval. *Neurocomputing*, 237:235–241.

Zha, Z.-J., Yang, L., Mei, T., Wang, M., and Wang, Z. (2009). Visual query suggestion. In *ACM international conference on Multimedia*, pages 15–24. ACM.

Zhi, T., Duan, L.-Y., Wang, Y., and Huang, T. (2016). Two-stage pooling of deep convolutional features for image retrieval. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 2465–2469. IEEE.