

Content-Based Image Retrieval Using Multiple Representations

Karin Kailing, Hans-Peter Kriegel, and Stefan Schönauer

University of Munich
Institute for Computer Science
{kailing,kriegel,schoenauer}@dbs.ifi.lmu.de

Abstract. Many different approaches for content-based image retrieval have been proposed in the literature. Successful approaches consider not only simple features like color, but also take the structural relationship between objects into account. In this paper we describe two models for image representation which integrate structural features and content features in a tree or a graph structure. The effectiveness of this two approaches is evaluated with real world data, using clustering as means for evaluation. Furthermore, we show that combining those two models can further enhance the retrieval accuracy.

1 Introduction

A common approach to model image data is to extract a vector of features from each image in the database (e.g. a color histogram) and then use the Euclidean distance between those feature vectors as similarity measure for images. But the effectiveness of this approach is highly dependent on the quality of the feature transformation. Often it is necessary to extract many features from the database objects in order to describe them sufficiently, which results in very high-dimensional feature vectors. Those extremely high-dimensional feature vectors cause many problems commonly described by the term 'curse of dimensionality'.

Especially for image data, the additional problem arises how to include the structural information contained in an image into the feature vector. As the structure of an image cannot be modeled by a low-dimensional feature vector, the dimensionality problem gets even worse. A way out of this dilemma is to model images with structured data types like graphs or trees. In this paper, we present two such approaches which take structural as well as content information into account. We also describe how the data mining technique of clustering can be used to evaluate the effectiveness of similarity models. Furthermore we show that the combination of the two models is able to enhance the accuracy of image retrieval.

2 Related Work

Numerous approaches for content-based image retrieval have been proposed in the literature. They are based on features like color [1], shape [2] or texture [3]. In [4] a graph-based approach similar to the one described in section 3.2 is used, while in [5]

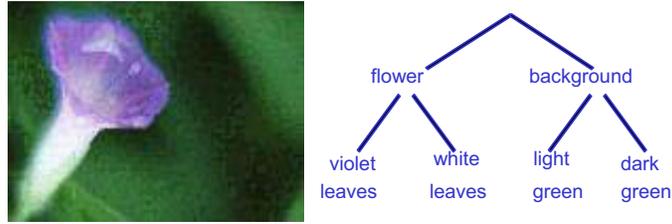


Fig. 1. An image and its inherent structure.

an edit distance measure is used to measure similarity of topological arrangements. [6] gives a nice overview of the different approaches.

However, all those approaches concentrate on single image features like color or texture. Our method differs from those proposed in the literature as we combine different approaches in order to achieve more meaningful results.

3 Content-Based Image Retrieval

In content-based image retrieval the use of simple features like color, shape or texture is not sufficient. Instead, the ultimate goal is to capture the content of an image via extracting the objects of the image. Usually images contain an inherent structure which may be hierarchical. An example can be seen in figure 1. In the following, we describe two models for image representation and similarity measurement, which take structural as well as content features like color into account.

3.1 Image Representation as Containment Trees

One way to model images for content-based retrieval is the use of trees representing the structural and content information of the images. In this section, we describe, how the structure of images can be extracted automatically based on the color of its segments. Additionally we show how the similarity between two such trees can be measured.

Transforming an Image into a Containment Tree To utilize the inherent structure of images for content-based retrieval, we model them as so called containment trees. Containment trees model the hierarchical containment of image regions within others.

To extract the containment tree of an image we first segment the image based on the colors of the regions using a region growing algorithm. The resulting segments are attributed with their color and size relative to the complete image. In a second step, the containment hierarchy is extracted from the set of segments by determining which regions are completely contained in other regions. In this context, a region R_{in} is said to be contained in a region R_{cont} if for every point $p \in R_{in}$ and every straight line $L \ni p$ there exist two points $o_1, o_2 \in R_{cont}$ with $o_1, o_2 \in L$ and o_1, o_2 are on opposite sides of p .

Measuring the distance between two Containment Trees To measure the similarity of containment trees, special similarity measures for attributed trees are necessary. A successful similarity measure for attributed trees is the edit distance. Well known from string matching [7, 8], the edit distance is the minimal number of edit operations necessary to transform one tree into the other. The basic form allows two edit operations, i.e. the insertion and the deletion of a node. In the case of attributed nodes the change of a node label is introduced as a third basic operation. A great advantage of using the edit distance as a similarity measure is that along with the distance value, a mapping between the nodes in the two trees is provided in terms of the edit sequence. The mapping can be visualized and can serve as an explanation of the similarity distance to the user.

However, as the computation of the edit-distance is NP-complete [9], constrained edit distances like the degree-2 edit distance [10] have been introduced. They were successfully applied to trees for web site analysis [11], structural similarity of XML documents [12], shape recognition [13] or chemical substructure search [11].

Definition 1 (degree-2 edit distance). *The edit distance between two trees t_1 and t_2 , $ED_2(t_1, t_2)$, is the minimum cost of all degree-2 edit sequences that transform t_1 into t_2 or vice versa. A degree-2 edit sequence consists only of insertions or deletions of nodes n with $\text{degree}(n) \leq 2$, or of relabelings:*

$$ED_2(t_1, t_2) = \min\{c(S) | S \text{ is a degree-2 edit sequence transforming } t_1 \text{ into } t_2\}$$

The main idea behind this distance measure is that only insertions or deletions of nodes with a maximum number of two neighbors are allowed.

Efficient Similarity Search for Containment Trees While yielding good results, the degree-2 edit distance is still computationally complex and, therefore, of limited benefit for searching or clustering in large databases. In [14], a filter and refinement architecture for the degree-2 edit distance is presented to overcome this problem. A set of new filter methods for structural and for content-based information as well as ways to flexibly combine different filter criteria are presented.

3.2 Image Representation as Segmentation Graphs

Graphs are another way to model images for content-based similarity search. They were successfully used for shape retrieval [15], object recognition [16] or face recognition [17]. In this section, we describe a content-based image retrieval system based on graphs which are extracted from images in a similar way as the trees in the preceding section.

Transforming an Image into a Segmentation Graph To extract graphs from the images, they are segmented with a region growing technique and neighboring segments are connected by edges to represent the neighboring relationship. Each segment is assigned four attribute values, which are the size, the height and width of the bounding box and the color of the segment. The values of the first three attributes are expressed as a percentage relative to the image size, height and width in order to make the measure invariant to scaling.

Measuring the distance between two Segmentation Graphs Most known similarity measures for attributed graphs are either limited to a special type of graph or are computationally extremely complex, i.e. NP-complete. Therefore they are unsuitable for searching or clustering large collections. In [18], the authors present a new similarity measure for attributed graphs, called edge matching distance.

Definition 2 (edge matching distance). Let $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ be two attributed graphs. Without loss of generality, we assume that $|E_1| \geq |E_2|$. The complete bipartite graph $G_{em}(V_{em} = E_1 \cup E_2 \cup \Delta, E_1 \times (E_2 \cup \Delta))$, where Δ represents an empty dummy edge, is called the edge matching graph of G_1 and G_2 . An edge matching between G_1 and G_2 is defined as a maximal matching in G_{em} . Let there be a non-negative metric cost function $c : E_1 \times (E_2 \cup \Delta) \rightarrow \mathbb{R}_0^+$. The edge matching distance between G_1 and G_2 , denoted by $d_{match}(G_1, G_2)$, is defined as the cost of the minimum-weight edge matching between G_1 and G_2 with respect to the cost function c .

The authors demonstrate that the edge matching distance is a meaningful similarity measure for attributed graphs and that it enables efficient clustering of structured data.

Efficient Similarity Search for Segmentation Graphs In [18] there is also a filter-refinement architecture and an accompanying set of filter methods presented to reduce the number of necessary distance calculations during similarity search. We employ the same approach to ensure efficient query processing in our experiments.

4 Evaluating the Effectivity of Similarity Models using Clustering

In general, similarity models can be evaluated by computing k-nearest neighbor queries. A drawback of this evaluation approach is that the quality measure of the similarity model depends on the result of few similarity queries and, therefore, on the choice of the query objects. A model may perfectly reflect the intuitive similarity according to the chosen query object and would be evaluated as "good" although it produces disastrous results for other query objects. [19] shows that clustering is a better way to evaluate and compare several similarity models. Clustering groups a set of objects into classes where objects within one class are similar and objects of different classes are dissimilar to each other. The result can be used to evaluate which model is best suited for which kind of objects.

5 Combining Multiple Representations for Clustering

Additionally, we can combine different similarity models to produce a better clustering result. Traditional clustering algorithms are based on one representation space. However, for complex objects often multiple representations exist for each object as in our case two different representations for each image. In [20], an efficient density-based approach to cluster such multi-represented data, taking all available representations into account, is presented. The authors propose two different techniques to combine the information of all available representations dependent on the application. The basic idea



Fig. 2. A typical cluster obtained with the graph model.

of this approach is to combine the information of all different representations as early as possible, i.e. during the run of the clustering algorithm, and as late as necessary, i.e. after using the different distance functions of each representation. To do so, the core object property proposed for the clustering algorithm DBSCAN [21] is adapted. Based on two input parameters (ϵ and k), DBSCAN defines dense regions (clusters) by means of core objects. An object $o \in DB$ is called *core object*, if its ϵ -neighborhood contains at least k objects. Usually clusters contain several core objects located inside a cluster and border objects located at the border of the cluster. In case of multiple representations the decision whether an object is a core object, is based on the local ϵ -neighborhoods of each representation and the results are combined to a global neighborhood. We argue that no single representation of an image models the intuitive notion of similar images adequately. All those different similarity models for image data have their own advantages and disadvantages. Therefore, clustering image data is a good example for the usefulness of the presented intersection-method. This method requires that a cluster should contain only objects which are similar according to all representations. Thus, it increases the cluster quality by finding purer clusters.

6 Experimental Evaluation

To test the effectiveness of the two presented approaches, we performed several clustering experiments. Our image database consisted of 1000 color TV-images which were segmented and transformed into trees and graphs in the way described in the sections 3.1 and 3.2. We clustered them using the DBSCAN algorithm [21] and the intersection algorithm presented in [20] to combine the two representations.

The results obtained with the two models separately were quite different. With the graph model we obtained several rather homogeneous clusters like the one depicted in figure 2 but also very diverse clusters like the one shown in figure 3. In general, it was possible to distinguish hockey images from the rest of the database rather well.



Fig. 3. A typical diverse cluster obtained with the graph model.

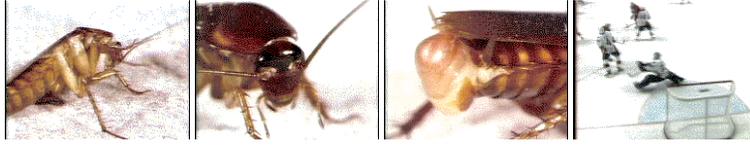


Fig. 4. A cluster of insects which could only be obtained with the combined model.

On the other hand, the use of the tree model only yielded one large and unspecific cluster and much noise. Obviously, this model alone is ill-suited for our image database.

But although the second model on its own did not yield any interesting results, the combination of both approaches turned out to be effective. Figures 4 and 5 show typical clusters obtained with the combination of the two models. As can be seen in figure 4, the combination yielded more homogeneous clusters as for example one of insect images. Those images belonged to a big and diverse cluster for the graph model. Additionally, the distinguishing power for the hockey images was preserved as shown in figure 5. In general, the clusters we obtained combining both representations were more accurate than the clusters we got using each representation separately. Obviously, the noise ratio increased if we combined the two representations.

7 Conclusions

In the diverse field of content-based image retrieval many different approaches have been proposed. In this paper, we described two models for image similarity which take into account structural as well as content information of an image. The presented models are based on tree and graph structures. With experiments on real-world data, we showed that the combination of those two approaches yields a performance gain concerning the specificity of the image retrieval process. This was done by means of clustering the images to compare the measures on a broad basis.

In our future work, we intend to combine the two presented similarity models with other representations like color histograms or text descriptions. Additionally, we want to further investigate and improve the query performance especially on very large image repositories.



Fig. 5. A cluster obtained with the combined model.

References

1. Flickner, M., Swahney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by image and video content: The QBIC system. *IEEE Computer* (1995)
2. Mehre, B., Kankanhalli, M., Lee, W.: Shape measures for content based image retrieval: A comparison. *Information Processing Management* **33** (1997) 319–337
3. Cullen, J., Hull, J., Hart, P.: Document image database retrieval and browsing using texture analysis. In: *Proc. 4th Int. Conf. Document Analysis and Recognition*. (1997) 718–721
4. Fuh, C.S., Cho, S.W., Essig, K.: Hierarchical color image region segmentation and shape extraction. *IEEE Transactions on Image Processing* **9** (2000) 156–163
5. Tagare, H., Vos, F., Jaffe, C., Duncan, J.: Arrangement - a spatial relation between parts for evaluating similarity of tomographic section. *IEEE Trans. PAMI* **17** (1995) 880–893
6. Smeulders, A.W.M., Worring, M., Santini, Simone Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. *IEEE Trans. PAMI* **22** (2000) 1349–1380
7. Levenshtein, V.: Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-Doklady* **10** (1966) 707–710
8. Wagner, R.A., Fisher, M.J.: The string-to-string correction problem. *Journal of the ACM* **21** (1974) 168–173
9. Zhang, K., Statman, R., Shasha, D.: On the editing distance between unordered labeled trees. *Information Processing Letters* **42** (1992) 133–139
10. Zhang, K., Wang, J., Shasha, D.: On the editing distance between undirected acyclic graphs. *International Journal of Foundations of Computer Science* **7** (1996) 43–57
11. Wang, J.T.L., Zhang, K., Chang, G., Shasha, D.: Finding approximate patterns in undirected acyclic graphs. *Pattern Recognition* **35** (2002) 473–483
12. Nierman, A., Jagadish, H.V.: Evaluating structural similarity in XML documents. In: *Proc. 5th Int. Workshop on the Web and Databases (WebDB 2002)*, Madison, Wisconsin, USA. (2002) 61–66
13. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing shock graphs. In: *Proc. 8th Int. Conf. on Computer Vision (ICCV'01)*, Vancouver, BC, Canada. Volume 1. (2001) 755–762
14. Kailing, K., Kriegel, H.P., Schönauer, S., Seidl, T.: Efficient similarity search for hierarchical data in large databases. In: *Proc. 9th Int. Conf. on Extending Database Technology (EDBT 2004)*. (2004) 676–693
15. Huet, B., Cross, A., Hancock, E.: Shape retrieval by inexact graph matching. In: *Proc. IEEE Int. Conf. on Multimedia Computing Systems*. Volume 2. (1999) 40–44
16. Kubicka, E., Kubicki, G., Vakalis, I.: Using graph distance in object recognition. In: *Proc. ACM Computer Science Conference*. (1990) 43–48
17. Wiskott, L., Fellous, J.M., Krüger, N., von der Malsburg, C.: Face recognition by elastic bunch graph matching. *IEEE Trans. PAMI* **19** (1997) 775–779
18. Kriegel, H.P., Schönauer, S.: Similarity search in structured data. In: *Proc. 5th Int. Conf. DaWaK 2003*. Volume 2737 of LNCS. (2003) 309–319
19. Kriegel, H.P., Kröger, P., Mashaël, Z., Pfeifle, M., Pötke, M., Seidl, T.: "Effective Similarity Search on Voxelized CAD Objects". In: *Proc. 8th Int. Conf. on Database Systems for Advanced Applications (DASFAA'03)*, Kyoto, Japan. (2003)
20. Kailing, K., Kriegel, H.P., Pryakhin, A., Schubert, M.: Clustering multi-represented objects with noise. In: to appear in: *Proc. 8th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'04)*, Sydney, Australia. (2004)
21. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *2nd Int. Conf. KDD*. (1996) 226–231