Extracting Symbolic Descriptors for Interactive Object Retrieval

Jochen Wickel, Pablo Alvarado, and Karl-Friedrich Kraiss

Lehrstuhl für Technische Informatik, RWTH Aachen
http://www.techinfo.rwth-aachen.de/

Abstract. In visual object retrieval, there are always queries which cannot be processed successfully. In these cases, it is desirable to extract imprecise symbolic information. We present an architecture that extracts symbolic descriptors of the objects shown in the query image. The method is based on a combination of numeric feature extraction and classification. We describe some examples of descriptors and present first experimental results.

1 Introduction

An object retrieval system finds objects in its database that are most similar to the one shown in a given query image (Fig. 1). Unfortunately, no system so far can perform this task perfectly. Since the way these systems work differs much from the way humans recognize objects, they sometimes return results that are implausible to their users. It is therefore desirable to have a system work with descriptors that are closer to human thinking. For instance, we would expect a system to reply "I have no idea what this is, but it's red and shaped like a circle" when shown an unknown object. We have integrated a detector for these kinds of descriptors in the AXIOM system [9], a modular object retrieval system designed to interactively retrieve three-dimensional objects when given a query image.

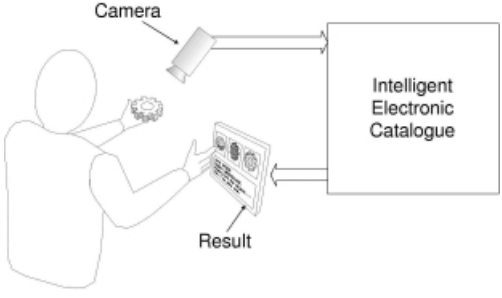

Fig. 1. Visual object retrieval. An image showing an object is presented to an intelligent catalogue which returns the desired information

J. Bigun and T. Gustavsson (Eds.): SCIA 2003, LNCS 2749, pp. 290–297, 2003.

We use a machine-learning approach to extract symbolic descriptors. The idea is that descriptors are exemplified by artificial prototype images. The descriptor detector can then be described as a recognition system that tries to detect the prototype in an image. There are two advantages of such an approach. First, is not required to hand-code the descriptor detection method, since they are determined by learning. Second, it allows the detection of descriptors that cannot be described by a simple representation in any low-level feature space.

The remainder of this paper is organized as follows. Section 2 highlights some related work. Section 3 explains the descriptor extraction architecture. In Section 4, we present some preliminary experimental results. Finally, Section 5 concludes the paper and presents ideas for future work.

2 Related work

There are two major areas where symbolic information is combined with image analysis. Examples for the first one, image annotation, are systems like Schema [4] or CITE [3]. They contain knowledge bases that enable a top-down verification of visual features determined from the image. Even though these systems are very powerful, they have to be adapted to each new scene type.

The second area is image and video retrieval. An example for relating images with textual labels is the system presented in [5]. The user can specify verbal attributes which are mapped into numeric constraints on low-level feature vectors. These constraints are used to retrieve video streams fitting them. The conversion from low-level feature to symbolic descriptors is similar to the one we present in the next section.

Image retrieval systems that support decision-making inevitably require some kind of symbolic output. For instance, the authors of [1] describe a visual inspection system that uses fuzzy rules for detecting surface defects. Other systems, like the content-based image retrieval system for neuroradiological images described in [7], extract symbolic descriptors that are interpreted by a subsequent specialized knowledge-based inference system.

A common drawback of all these methods is that they require training images from the same domain as the query images. To our knowledge, no method using artificial training data has been published yet.

3 Symbolic Descriptor Extraction

The goal of symbolic descriptor extraction is to derive a symbolic description of an object represented by a query image. The basic structure of the extractor is shown in Fig. 2: First, numerical feature vectors are extracted, which are afterwards subjected to a classification stage. The final symbolic descriptors result from a conversion of the classification results. These stages are described in the following paragraphs.

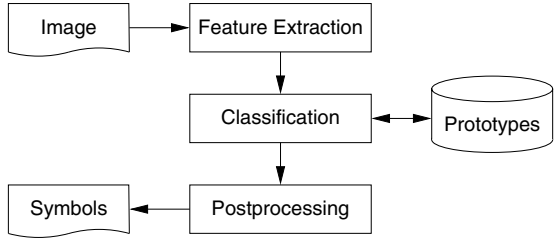

Fig. 2. The structure of the symbolic descriptor extractor

3.1 Primitive Feature Extraction

Given an image, the primitive feature extraction stage extracts a suitable numerical feature vector. The type of numerical feature used depends entirely on the desired symbolic descriptors. In the AXIOM system, there are many different kinds of feature extractors sensitive to color, shape and texture of objects [9]. Usually they are used for directly recognizing objects, but some of them are also suited to serve as a basis for computing symbolic descriptors.

An interesting example are two shape features found in the MPEG-7 specification: The Curvature Scale Space method [8] and the region-based shape feature [2]. The first method requires the contour of an object while the second one uses a background-object mask, therefore they have to be preceded by an image segmentation stage.

Both types of features show properties that make them suitable for object retrieval: The region shape feature vector is invariant against scaling and in-plane rotations. The CSS feature is scaling invariant, but, in its pure form, not rotation invariant. However, the latter invariance can be achieved by considering only the width and height of the most characteristic blobs in the CSS image, ignoring their absolute positions. The implementation we used can be found in the computer vision library LTI-Lib (http://ltilib.sourceforge.net/).

3.2 Prototype detection

The feature vectors that are delivered by the first stage are passed to a classification module. In training mode, the classifier is trained with features extracted from distorted prototype images. In classification mode, it is fed the feature vectors extracted from images that are to be described. The type of classifier primarily depends on the complexity and the nature of the extracted primitive features. The modularity of the system allows for different classifiers to be used for different kinds of low-level features. This is an important property because, even though a broad range of problems can be covered by k-nearest neighbor or maximum likelihood methods, all classification methods are usually well-suited for some kinds of data sets and ill-suited for others.

Fig. 3. The primitives used for the experiments: a-shaped, bar, circle, ellipse, hexagon, l-shaped (top row); pentagon, rectangle, square, t-shaped, triangle, u-shaped (bottom row)

3.3 Conversion

In the AXIOM system, all classifiers yield an a-posteriori distribution $P(C|X)$ of an observation X belonging to a class C. This distribution has to be converted into a symbolic descriptor or a vector of symbols. We define two interpretations of a classification result: The first one is a multi-valued discrete random variable S, each value representing a primitive object. We define $P(S = s_i|X) = P(C = c_i|X)$. The extracted symbolic descriptor for an image is then drawn from this distribution.

The second interpretation is a vector of binary random variables S_i, each one denoting the presence of shape s_i in an image. Then, the probability distribution is defined analogously with $P(S_i|X) = P(C = c_i|X)$.

4 Experimental Results

For a first assessment of the usefulness of the descriptors described in the previous section, we defined an extractor using the MPEG region shape features [2] and a k-nearest neighbor classifier ($k = 10$). This choice proved to be the most promising combination in preliminary experiments. The shapes were determined by the primitives shown in Fig. 3.

In order to "train" the classifier, we generated distorted images of the primitives by rotating them in 3D space around a tumbling axis (out-of-plane). This resulted in a training set of 56 images per primitive.

The symbolic descriptor extractor was then used to extract shape labels on two sets of images, both from a 3D object recognition task [9]. Both sets contain images of the same 36 objects, but are created in different ways (see below for details).

In order to assess the quality of the derived descriptors, we subjected them to a statistical evaluation. For each object o, the probabilities $P(S_i|o)$ of a 3D object looking like shape S_i is computed as the average of the probabilities for each image showing this object. The result can be interpreted as a 12-dimensional random variable. In case of an optimal descriptor assignment, the correlation coefficients should not be larger than zero.

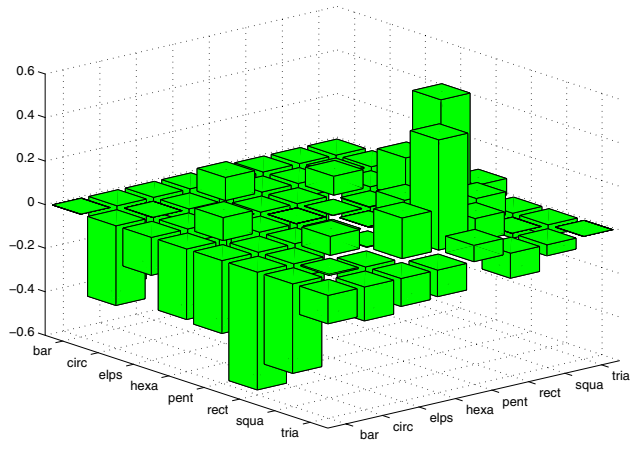

Fig. 4. A plot of the correlation coefficients of the assigned label probabilities for CAD images. For readability, the main diagonal was set to 0 and only the eight most frequently assigned labels are shown

4.1 Experiment 1: CAD images

The first set of images was obtained by rendering 36 3D CAD models, each one viewed from 216 viewpoints. The test set thus consisted of over 7500 images. For these images, the segmentation process could always be performed perfectly.

As can be seen in Fig. 4, which shows the correlation coefficients for the eight shapes assigned most often, the coefficients are less than zero for most entries, meaning that the descriptors in fact react to different shapes. The most notable exception are the labels "pentagon" and "square" which are highly correlated.

One reason for that behavior is that the objects were shown from different viewpoints, and thus one object can assume different shapes. In Fig. 5(a), one

Fig. 5. (a) different views of the object 31010, one appearing as pentagon, the other one as square; (b) different views of object 31019 with the assigned symbolic shape descriptors

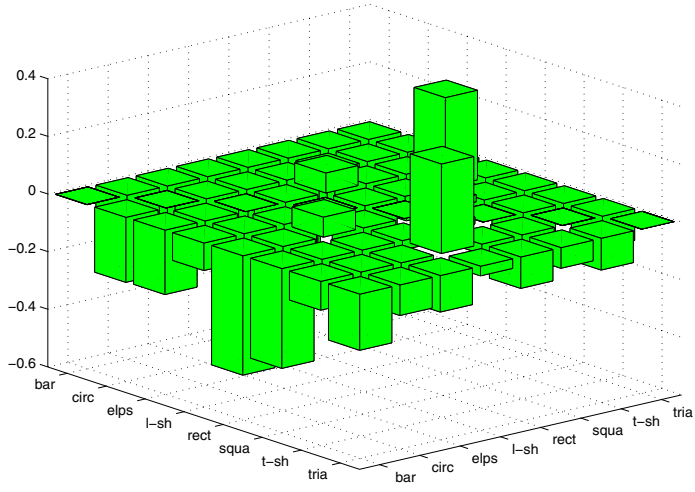

Fig. 6. A plot of the correlation coefficients of the assigned label probabilities for camera images. For readability, the main diagonal was set to 0 and only the eight most frequently assigned labels are shown

can see that this objects looks like a pentagon in some views, whereas it looks like a square in others. This is correctly reflected in the assigned shape labels.

Figure 5(b) shows another example of labeled object images. The object, a disc, is assigned the label "bar" when viewed from the side and "circle" for a front view. The anticorrelation of the "bar" label and any other label can be explained by the fact that many of the shown objects are bars and thus are assigned the "bar" label in all views, whereas the other objects receive this label only for a few degenerate views.

4.2 Experiment 2: Camera images

The second set of images contained 40 real camera images for each object (1440 images in total). Due to errors in the images, segmentation could not always be performed successfully. Thus, the symbolic feature was computed 1412 images, the remaining 28 were rejected. The correlation coefficients of the resulting distribution are shown in Fig. 6.

The main difference is in the set of shapes that are assigned most frequently. The "pentagon" and "hexagon" labels found in the first experiment have been replaced by "l-shaped" and "t-shaped". The most likely explanation for this are the statistical variations in the data set. However, the basic property of the data remain the same as in the CAD image case: Only descriptors that may be generated by different views of the same object are correlated.

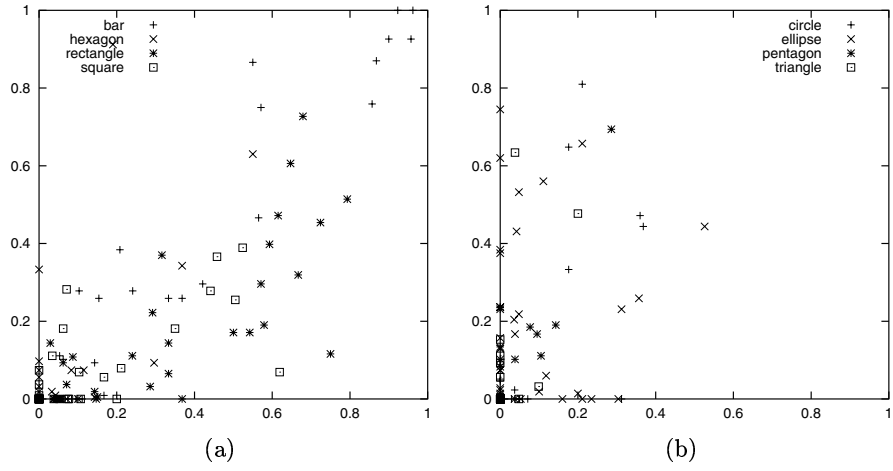

Fig. 7. Scatter plots of the probabilities for symbolic shape descriptors assigned by humans (x-axis) vs. ones extracted by our system (y-axis). (a) well correlated features; (b) badly correlated features

4.3 Experiment 3: Comparison with human labelings

In order to test the correspondence of the extracted symbolic descriptors with human terms, a Web-based recognition experiment was set up, presenting the subjects with a randomly selected object masks. These masks were generated by segmentation of the CAD images from the first experiment. The subjects then chose the shape that they found to resemble the shape of the object most closely. This resulted in a set of about 1000 ratings. From this data, the probability $P(S = s_i|o)$ of an object having shape s_i was computed, just as for the extracted descriptors. This resulted in a pair of probabilities for each shape. Figure 7 shows scatter plots of these pairs, where each point corresponds to the probability of the shape given one object. The figure shows the same descriptors as shown in Fig. 4. The left plot shows those features that correspond well with human ratings. The right one shows features which do not correspond well.

An interesting property of the right plot is that the choices of the human test subjects consistently result in a lower probability than those assigned by the descriptor extractor. A possible reason for this property might be that the human subjects had preferences for certain shapes when there was no exact match. An example for this may be seen for the shapes "rectangle" (Fig. 7a) and "pentagon" (Fig. 7b): While humans rated rectangles as more probable as the extractor did, this relation is reversed for pentagons.

As a consequence, there is a promising agreement with human judgement for some of the descriptors, but for others, the results differ. Even though there are some plausible explanations for this circumstance, the actual cause remains to be investigated.

5 Conclusions and future work

We have presented a generic framework for extracting symbolic descriptors from images in an interactive object retrieval application. In a preliminary evaluation, the descriptors have shown to be both specific and robust. Therefore, the modular prototype-based approach presented seems a promising tool for deriving descriptions usable for human-computer interaction. However, a thorough quantitative analysis of the descriptors has yet to be performed. Specifically, we plan to further investigate the relationship between the shape labels extracted by our system and those assigned by humans.

We also plan to evaluate the usefulness for the extracted descriptors to enhance recognition quality. For 3D object retrieval, it seems plausible to make use of the 3D information available in the reference data. We therefore aim to use the symbolic descriptors for 3D shapes defined in [6] for improving the classification results obtained by the view-based recognition process.

Acknowledgements

This project has been funded by the Heinz-Nixdorf Foundation. The test objects were kindly provided by fischerwerke Artur Fischer GmbH & Co. KG.

References

[1] M. Bariani, R. Cucchiara, M. Piccardi, and P. Mello. Data mining for automated visual inspection. In *Proceedings of First Int. Conf. On Practical Application of Knowledge Discovery and Data Mining (PADD '97)*, pages 51–64, 1997.

[2] M. Bober. MPEG-7 visual shape descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):716–719, June 2001.

[3] C. Dillon and T. Caelli. Learning image annotation: The CITE system. *Videre*, 1(2):90–121, 1998.

[4] B. Draper, R. Collins, J. Brolio, A. Hanson, and E. Riseman. The Schema system. *Int. Journal of Computer Vision*, 2(3):209–250, January 1989.

[5] S. Hollfelder, A. Everts, and U. Thiel. Concept-based browsing in video libraries. In *Proc. of the IEEE Forum on Research and Technology Advances in Digital Libraries (IEEE ADL 99)*, pages 105–115, 1999.

[6] T. Krüger, J. Wickel, P. Alvarado, and K.-F. Kraiss. Feature extraction from VRML models for view-based object recognition. In *Proc. of the 4th European Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2003)*, pages 391–394, 2003.

[7] Y. Liu, W. E. Rothfus, and T. Kanade. Content-based 3d neuroradiologic image retrieval: Preliminary results. In *IEEE Int. Workshop on Content-based Access of Image and Video Databases*, pages 91–100, Jan. 1998.

[8] F. Mokhtarian, S. Abbasi, and J. Kittler. Robust and efficient shape indexing through curvature scale space. In *Proc. of British Machine Vision Conf.*, pages 53–62, Edinburgh, UK, 1996.

[9] J. Wickel, P. Alvarado, P. Dörfler, T. Krüger, and K.-F. Kraiss. Axiom — a modular visual object retrieval system. In M. Jarke, J. Koehler, and G. Lakemeyer, editors, *KI 2002: Advances in Artificial Intelligence*, pages 253–267. Springer, 2002.