16^{th} *Computer Vision Winter Workshop*
Andreas Wendel, Sabine Sternig, Martin Godec (eds.)
Mitterberg, Austria, February 2-4, 2011

Image retrieval by shape-focused sketching of objects

Hayko Riemenschneider, Michael Donoser, Horst Bischof
Institute for Computer Graphics and Vision,
Graz University of Technology, Austria
{hayko, donoser, bischof}@icg.tugraz.at

Abstract. *Content-based image retrieval deals with retrieval in large databases using the actual visual content. In this paper we propose to use hand-drawn object sketches highlighting the outline of an object of interest as query. Due to the lack of appearance, the focus lies on the shape of an object. Such a scenario requires a common representation for the sketch and the images. We propose novel shape-based descriptors that are calculated on local contour fragments. The contour descriptors are stored in a hierarchical data structure, which enables efficient retrieval in sub-linear time, potentially handles millions of images, and does not require retraining when inserting new images. We demonstrate superior performance in this query-by-shape-sketch retrieval for our novel features, and efficient retrieval in 50 milliseconds on a standard single core computer.*

1. Introduction

Image retrieval [5, 14, 15], which deals with the finding of similar images to a given query in large databases, has seen tremendous progress in the last years. Impressive advances were achieved in terms of number of images indexed in the database (up to millions) [5, 19, 23], types of features able to process (color, texture, shape) [11, 12] and most recently also the types of input. The last part deals with what kind of input is provided as query to run image retrieval, for example semantic language based queries, full feature images, or scene and object sketches [7, 13].

In general, mainly three different approaches of how to define the query in a retrieval system can be distinguished. The first group extends standard text retrieval systems relating them to images. The second group considers fully featured images as query, which contain rich scene information in appearance and shape. However, concerning a user guided image retrieval system, such data may not be available, because the user looks for a specific type of image and cannot provide an exemplar image, since this is the actual goal of the search. The third approach uses hand-drawn sketches, showing the desired scene colors or shape of objects, where the visual similarity is defined on a more abstract semantic level.

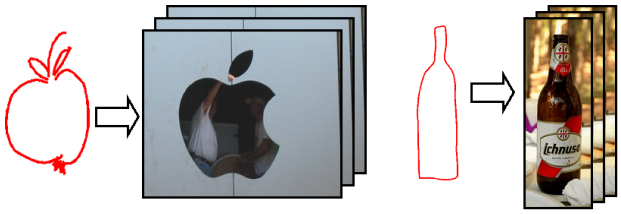

Figure 1. Query by shape sketch image retrieval: Sketching an object outline is the most intuitive user input to support visual image search. Contrary to scene sketching with focus on appearance, the main issue for this novel approach is efficient matching of the shape of an object as well its discrimination to background clutter.

The goal of this paper is to introduce a content-aware image retrieval system, which solely uses a sketch of the outline of an object as query as it is illustrated in Figure 1. This enables a novel intuitive system, where users simply sketch an object of interest on e. g. a tablet PC and immediately retrieve images containing the specified object.

Our content-based image retrieval system is based on a novel feature for describing the local shape of contour fragments and an efficient data structure to retrieve images from large databases in short response time. Inherent properties of our system are the focus on shape, efficient fragment matching considering connectedness of sketch stroke sequences and possible handling of occlusions. We demonstrate how our shape descriptor improves retrieval performance and allows for a content-based image retrieval focused on objects rather than scenes.

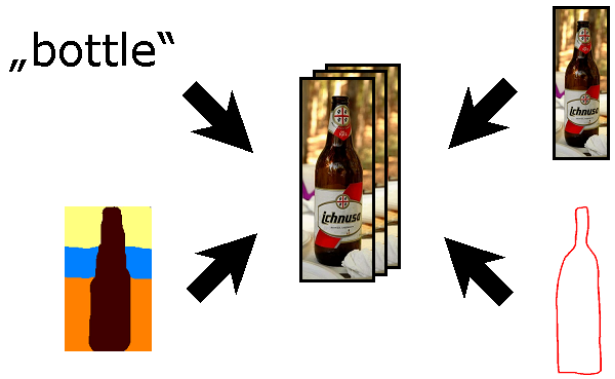

Figure 2. Overview of content-based image retrieval by query type: text, image, scene- and shape-sketch.

2. Related work

Methods for content-based image retrieval can be classified into the following four fields: (a) semantic language-based , (b) image-based, (c) scene sketch-based and (d) object sketch-based, see Figure 2 for an overview. In the following sections, related work in these four fields and properties are discussed.

2.1. Query by text (language-based)

The first field, denoted as *query-by-text*, deals with retrieving images by matching the user input to meta data provided with images, as for example is available on websites or image annotation databases. The features analyzed are not considering image content but rather text describing the content. Powerful scoring functions have been developed for such text retrieval systems to accurately measure the similarity between language data like the well-known term frequency / inverse document frequency (TF/IDF) scheme. Modern search engines such as Bing or Google deal only with semantic queries, whereas for example Cortina [11][1] is a combination of semantic knowledge and image features from the MPEG-7 specification [22] and the SIFT descriptor [17].

2.2. Query by image (full feature images)

In the second field, denoted as *query-by-image*, a single image is provided as query and the most similar images from the database should be obtained. This is for example required for re-localization in 3D reconstruction methods or to identify near duplicate images for copyright protection. The key features are extracted from the full extend of visual information in terms of texture, color and shape.

For example, TinEye[2] creates a unique fingerprint for a complete image (actual technique not revealed) to find the exact matches including crops, editing and resizing. Windsurf[3] retrieves images based on wavelet-indexing of images under region fragmentation. That is, multiple region segmentations are described and used in a one-to-many matching setup [1]. CIRES[4] uses perceptual grouping on low-level edges to obtain a structure of the image, which is a high-level semantic cue for retrieval together with features from Gabor filters and Lab color space [12]. FIDS[5] focuses on efficient retrieval by using color, edge histograms as well as wavelet decomposition [3].

Most of these approaches focus on complete image retrieval, which given the full feature image as input delivers visually similar and even near-duplicate retrieval results. Recent *query-by-image* retrieval systems [5, 14, 18] deal with better scoring strategies and more effective vocabulary construction.

2.3. Query by scene sketch (color drawings)

The third field, denoted as *query-by-scene-sketch*, uses a manual drawing reflecting an image scene by color as query. The user provides a drawing, where complex visual features may not be used because there is simply no data on which to compute them since the sketch is more a cartoon-like drawing.

For example, Retrievr[6] extracts a multi-resolution wavelet fingerprint of the complete image comparing color and shape [13]. The compression to just 20 coefficients allows efficient retrieval.

2.4. Query by shape sketch (line drawings)

The last field of image retrieval systems, which we denote as *query-by-shape-sketch*, uses simple shape sketches as query. The user simply draws a rough outline of an object focusing entirely on the shape as it is illustrated in Figure 1. In our opinion this level of user interaction provides the most natural extension of a language based word-level query, since it enables intuitive systems, where users can simply sketch an object e. g. on a Tablet PC with only a small amount of user interaction required. Previous work in this field uses only histograms or full image similarities to retrieve images containing similar content.

[1] http://vision.ece.ucsb.edu/multimedia/cortina.shtml

[2] http://www.tineye.com

[3] http://www-db.deis.unibo.it/Windsurf/

[4] http://cires.matthewriley.com/

[5] http://www.cs.washington.edu/research/imagedatabase/demo/fids/

[6] http://labs.systemone.at/retrievr/

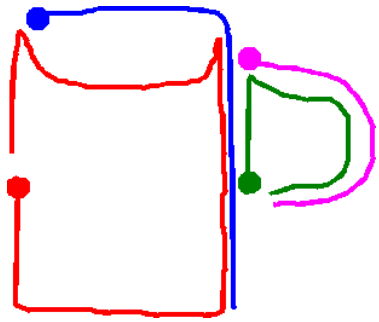

Figure 3. Properties of a user sketch: a) focus on shape, b) multiple strokes (shown in different colors), c) direction of drawing (shown by thicker starting point) and d) sequence information of the connected strokes.

We propose a novel scheme to combine sketches and shape cues using powerful local shape descriptors to retrieve images with similar objects.

3. Image retrieval by shape-sketch

The goal of this work is to enable efficient image retrieval in large databases based on modeling an object of interest using a sketch of the object shape. Hence, we define the term *sketch* as a thin line drawing by a user by means of an electronic pencil. As shown in Figure 3, such a sketch focuses on shape, has no appearance information, may contain one to many line strokes and has defined end points for each stroke (defining a valuable stroke point ordering).

In the following sections we will outline our novel retrieval method that uses such sketches as query to efficiently retrieve images containing the sketched object from potentially large databases. The core idea is to describe both the sketch and the images in terms of a bag of local fragment codewords, where codewords are fragment prototypes (found by comparing fragment shape) that are obtained from edges in the image database. For this we describe local fragments by a powerful shape descriptor that is explained in detail in Section 3.1. In Section 3.2 we describe how a hierarchical data structure denoted as *vocabulary tree* can be used to define our vocabulary of codewords. The vocabulary is built by analyzing the image database, nevertheless once the data structure is built, new images can be inserted without the need of re-training. Finally, in Section 3.3 we show how to use the obtained vocabulary tree in our *query-by-shape-sketch* object retrieval system.

3.1. Local contour fragment description

In a cluttered environment it is important to be able to discriminatively describe shape cues and distinguish them from mere background clutter. Additionally, for a content-based image retrieval system, efficient processing is a vital aspect. For this reason time-consuming learning tasks have to be moved to the offline preprocessing stage. Current state-of-the-art systems based on complex shape features still require a lot of online processing time and are dissimilar in terms of description of object sketches and images, i. e. they do not allow the same description. For this reason we made a thorough analysis of the related work in shape analysis focusing on speed and possible similar description of a binary shape and a full feature image. Possible descriptors include the Edge Histogram Descriptor (EHD) which is specified in the MPEG-7 standard [22], the Shape Context (SC) [2], the Turning Angle (TA), which is a subset of the Beam Angle Histograms (BAH) [20], and the PArtial Contour and Efficient Matching (PACEM) descriptor [21], which is a recent shape descriptor designed for partial matching and encoding of sequence information.

In this work we extend the shape description from [21] to enable efficient content-based image retrieval. As will be described in detail in the experimental section, our new shape descriptor makes *query-by-shape-sketch* feasible and successful because of an immense speedup and a powerful description of the shape of local fragments.

We define the term *contour* as a connected sequence of points, which might come from an edge obtained from an image or from a stroke from the input sketch. Further, a *contour fragment* is a connected subset of a contour. Essential to our description is that all contour fragments are an ordered list of points. Our descriptor is now calculated for such local contour fragments, all having a fixed number of points L and it considers the available ordering of the points. In comparison, the Shape Context (SC) [2] descriptor loses all the ordering information due to the histogram binning. It is further important to note, that the image edges and user strokes may be over-fragmented and broken into multiple contours. Contrary to [21], where partial matching is used to overcome this fragmentation, we simply analyze purely local contour fragments.

Our descriptor is inspired by the chord distribution. A chord is a line joining two points of a region boundary, and the distribution of their lengths and angles was used as shape descriptor before, as for example by Cootes et. al [6] or in the work on

Geometric Hashing [24]. Our descriptor analyzes these chords, but instead of building histograms of their distributions, we use the relative orientations between specifically chosen chords.

Our descriptor is based on angles α_{ij} which describe the relative spatial arrangement of the points $P_1 \dots P_L$ located on the analyzed contour fragment. An angle α_{ij} is calculated between a chord $\overline{P_i P_j}$ from a reference point P_i to another sampled point P_j and a chord $\overline{P_j P_\infty}$ from P_j to P_∞ by

$$\alpha_{ij} = \sphericalangle\left(\overline{P_i P_j}, \overline{P_j P_\infty}\right), \tag{1}$$

where $\sphericalangle(\dots)$ denotes the angle between the two chords and P_∞ is the point at vertical infinity. Thus the angle is calculated between the chord and a vertical line.

In the same manner L different angles $\alpha_{i1} \dots \alpha_{iL}$ can be calculated for one selected reference point P_i. Additionally, each of the sampled points can be chosen as reference point and therefore a $L \times L$ matrix A defined as

$$\mathrm{A} = \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1L} \\ \vdots & \ddots & \vdots \\ \alpha_{L1} & \cdots & \alpha_{LL} \end{pmatrix} \tag{2}$$

can be used to redundantly describe the entire shape of a fragment with length L. This descriptor matrix is not symmetric because it considers relative orientations. Please note, that such a shape descriptor includes local information (close to the main diagonal) and global information (further away from the diagonal) and it additionally encodes the global orientation of the fragment. In such a way the shape of every contour fragment of length L can be described by an $L \times L$ matrix.

3.2. Fragment vocabulary generation

In general, the goal of a content-based image retrieval system (CBIR) is to provide fast results on a large scale database. Most related work on *query-by-scene-sketch* and *query-by-shape-sketch* focuses on an approximated nearest neighbor search to achieve this. In this work we propose to use hierarchical data structures as introduced in *query-by-image* research to cluster and efficiently search our shape descriptors for defining a visual vocabulary. We apply a data structure known as *vocabulary tree* [19] for our purposes, which exhibits the benefits of data adaption and ability to handle high dimensional features as contrary to nearest neighbor search or kd-trees [10].

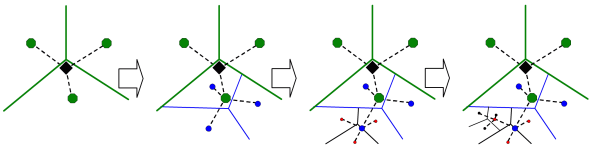

Figure 4. An example vocabulary tree for three cluster centers and a depth of four levels [19]. Each hierarchical level contains a part of the previous data and refines the clustering detail allowing for better data adaptation.

The vocabulary tree [19] is a highly effective way to define the vocabulary for bag-of-word representations and assign query descriptors to the codewords. The approach uses k-means clustering for each level of the tree. This yields a hierarchy of clusters which again is used to efficiently traverse the vocabulary tree and find matching cluster centers.

In its definition a vocabulary tree is a data structure of k cluster centers and a depth of l levels. Figure 4 shows an illustration from Nister and Stewenius of a vocabulary tree built for three cluster centers and a depth of four levels. For each new level the data clustered to the number of centers and divided. A new level of clustering provides more detailed quantization of the descriptors.

The cluster centers are referred to as nodes of the tree and the nodes at the last level are known as leaves. Each of these nodes contains an inverted file list. This list maintains an index to the images whose feature descriptors are included in the respective nodes. So instead of holding the actual descriptors themselves, only a correspondence between best matching node and image identifier is available.

Further each node contains a weight based on entropy. The more images are included in a node the less distinctive it becomes. Nister and Stewenius define various voting strategies for retrieval. First, the *flat strategy* defines a scoring where only the leaf nodes are used. If a descriptor of an image matches to a node in the lowest level, its weight is included in a sum later normalized by the number of descriptors in total. Second, the *hierarchical strategies* define scoring based on how many levels upwards from the leaf level are also considered during scoring. While the second one improves the recognition rate, the *flat scoring* allows much faster retrieval. We adopt this strategy and define the weight w_i of a node as

$$w_i = ln(\frac{N}{n_i}), \tag{3}$$

where the total number of images N in the vocab-

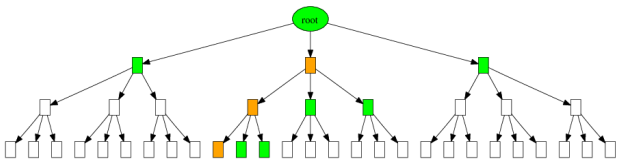

Figure 5. Illustration of the hierarchical traversal through a vocabulary tree: Only the nodes in green are considered at each level and the orange nodes indicate the best matches. These determine the k of k^{l+1} cluster nodes which are considered at the next level $l+1$.

ulary tree and the number of images n_i which are contained in a node i are used as entropy measure.

The final score s is determined by the sum over all nodes where the query descriptor matches this node. The frequency of matches for each descriptor is used and normalized by the total number of descriptors – for the query image and the already known images in the vocabulary tree. The final score is then defined as

$$s = \sum_i \frac{w_i \times q_i \times d_i}{Q \times D} \tag{4}$$

where w_i represents the weight of the current node, q_i and d_i the number of times a descriptor for a query or database image passed through the current node, and Q and D are the total number of query or database descriptors respectively.

We use the vocabulary tree to create a general shape vocabulary for an efficient retrieval system. For this, we extract Canny edges [4] from the images of our database using the standard hysteresis thresholding in Matlab and link the results to a set of coordinate lists. This linking is done by analyzing an 8 connected neighborhood [16]. Each edge is represented as an ordered sequence of sampled points and we extract heavily overlapping contour fragments from every edge (each of length L), by one point shifting, so that neighboring contour fragments overlap by $L-1$ points. The fragment length L is selected as a balance between discriminative power and edge fragmentation. For each fragment individually we sort the sequence clockwise and calculate the corresponding shape descriptor matrix of size $L \times L$ as it is described in Section 3.1. All obtained descriptors are then used to build the vocabulary tree. In such a way, at the final level of the clustering, the leaves define the desired codewords representing prototypes of contour fragments.

Once the vocabulary tree is generated, it may be used for fast codeword assignment and scoring, as well as insertion of new images for retrieval. For each of its description vectors, the top nodes and their cluster centers are matched. Only the children of the best matched cluster center are then matched again. This reduction of search space allows for a complete search of the vocabulary in $k \times l$ comparisons. Thus for a structure of ten cluster centers and six levels searching the one million leaf nodes for a best match only requires 60 comparisons, see Figure 5 for an illustration of this process.

During the insertion of a new image this advantage is used to find the best matching leaf node quickly. For each of the shape descriptors such a match is sought. Then, a new image identifier is included into the nodes' inverted file lists and their weights are updated. No further steps are required. The same hierarchical matching is used to determine the best matching leaf nodes for retrieval.

3.3. Retrieval system

For retrieving images from a database a user has to provide a shape sketch by drawing strokes. Therefore, we created a sketching interface, where the user draws on a tablet computer with an electronic pen or mouse. This directly allows to store the location and sequence of strokes. In such a way our object model has the following four attributes (refer to Figure 3): The sketch models the shape of the object of interest the user is searching for. It contains multiple strokes, which are the user-drawn lines to outline the object. Each stroke has a direction of drawing given by the user. Finally, adopted from the direction, each stroke has a sequence in which points along the stroke may be sampled.

To be able to retrieve images from the database, each stroke is modeled as a contour and highly overlapping contour fragments, as described in Section 3.2, are extracted for every stroke. All contour fragments are represented by our proposed powerful shape descriptor and the descriptors are passed to the vocabulary tree to obtain corresponding codewords. The retrieval result is a ranking of all images in the database using the final similarity scores as defined in Equation 4.

4. Experiments

The goal of this work is to propose a novel structure-capturing shape cue for content-based image retrieval (CBIR) systems. For this reason, we focus on evaluating shape cues for the quality in image retrieval. Due to the design of our retrieval system,

it shows the following properties. The shape vocabulary generation is performed in an offline stage and stays the same over all experiments. The vocabulary tree allows a fast retrieval of images as well as insertion of new images in constant time. The computation only depends on the number of k clusters and l levels chosen for the vocabulary, which is k = 3 and l = 6 for all our experiments. The constant time for insertion or retrieval of new images is thus $O(k \times l)$, which is (almost) independent of the number of images in the database. Since we focus on efficient local shape features, the time for calculating the descriptors is a few milliseconds. The full retrieval is performed on average in 50 milliseconds seconds per object sketch.

4.1. Shape-based features

For evaluation of our contour descriptor, we analyzed four additional descriptor methods. The Shape Context (SC) [2] is a correlated histogram of edges and is intended to provide a description for a set of points to determine their correspondences. The description is a normalized binned histogram, however in a log-polar layout to capture the relative distribution of points. The Turning Angle (TA) is a subset of the Beam Angle Histogram (BAH) [20]. The BAH is a histogram over beam angle statistics, where the beam angles θ_{ij}, at points on the shape P_i, i = 1, 2, . . . , are the chord lines (P_{i-j}, P_i) and (P_i, P_{i+j}). The PArtial Contour and Efficient Matching (PACEM) [21] is a recent shape descriptor designed for partial matching and encoding of sequence information.

4.2. Experimental setup

The experiment is designed to evaluate the performance of a *query-by-shape-sketch*, where rich visual features are not available. As it is difficult to evaluate an interactive user scenario, we setup the experiment to use the ETHZ shape classes [9] of 255 images containing five classes and let several users draw sketches for each class. The benefits are that the class for each image is known and we can use it to evaluate the retrieval performance, which is otherwise not well-defined in large image retrieval systems, where the exact number of true positive matches is not known.

For evaluation we use all obtained sketches, which represent the range of variations of typical user sketches. See Figure 6 for an overview of some of the sketches, which are provided as query input to the image retrieval system. This new sketch dataset contains 700 sketches drawn by 36 users. There are on average three user strokes with a length of 320 pixels. We use contour fragments of length 100 and sample every 5th point, leading to a length L = 20, which in experiments showed is a reasonable balance between discriminative power, dimensionality and limitations due to edge fragmentation.

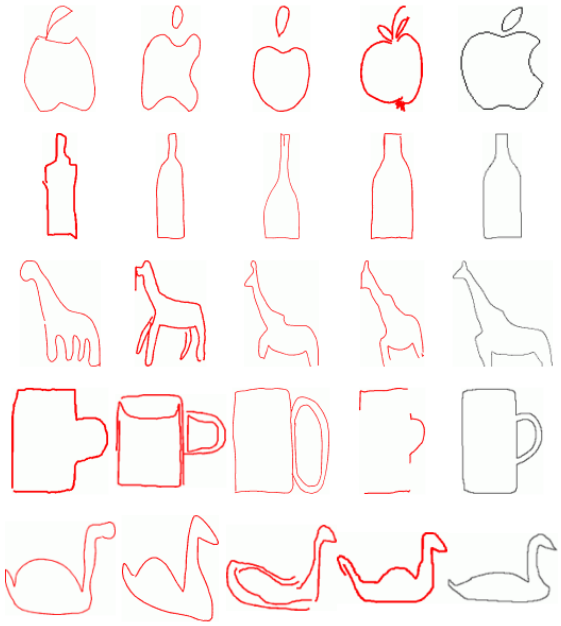

Figure 6. Subset of the 700 user sketches for the five ETHZ classes used in evaluation. The sketches cover the range of user input in a *query-by-shape-sketch* retrieval system. Second last column shows top performing sketches, and right column the sketches by Ferrari [9].

For this dataset the performance measure is the top-T ranked results, where the top-T score is defined as the number of true positive images (ground truth class vs. sketch query class) over the top T = 20 result images. This performance score shows how many retrieved images actually contain the desired object.

4.3. Results and discussion

Table 1 shows a summary of the average results of the 700 queries for the top-20 ranked images. The results show that the performance scores of the novel *query-by-shape-sketch* image retrieval paradigm are still moderate, however clearly demonstrate the benefits of using a shape descriptor, which captures the sequence of user strokes. Our descriptor performs on average 25% better than other shape descriptions.

Figure 8 shows a recall plot for the retrieval task, where the number of top ranked images was varied

Method	Sketch (avg./best)		Ferrari [9]
SC [2]	23.5%	41%	20%
TA	20.7%	44%	20%
BAH [20]	19.6%	55%	24%
PACEM [21]	19.2%	50%	20%
Proposed	**48.5%**	**87%**	**58%**

Table 1. Percentage of true positives within first 20 retrieved images using each of the 700 sketches of the novel dataset (average and best results) and the hand-drawn prototype models (right column).

from T = 1 to T = 20. The retrieval score is consistent over all top ranked images.

For completeness, we can evaluate the individual class results of the ETHZ dataset. This is not relevant for the retrieval systems, however the confusion table in Figure 7 shows that some categories can be modeled better than others. The average percentage of true positives within the first 20 retrieved images over all user sketches (including very crude ones) is for Applelogo 59%, Bottle 57%, Giraffe 66%, Mug 38%, Swan 23%. This distribution of performance is also visible when using the hand-drawn prototype models provided by Ferrari et. al [9]: Applelogo 60%, Bottle 85%, Giraffe 90%, Mug 20%, Swan 35%. Furthermore the scores if only considering the top performing sketch per class yields: Applelogo 100%, Bottle 90%, Giraffe 100%, Mug 80%, Swan 70%. The classes for swans and mugs are the hardest, since they are most often confused with applelogos and bottles, respectively, due to similar local shapes (head, neck and straight vertical lines).

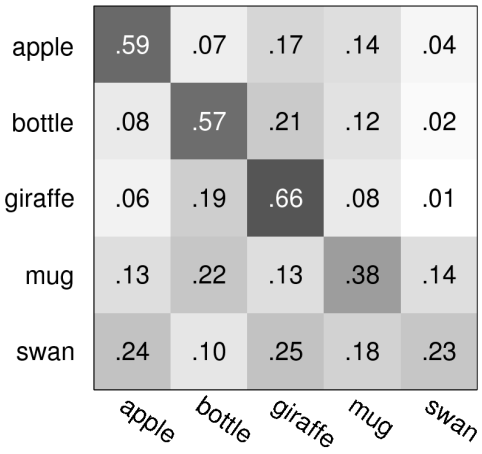

Figure 7. Confusion table for average scores for each ETHZ shape class [9] on the 700 user sketches.

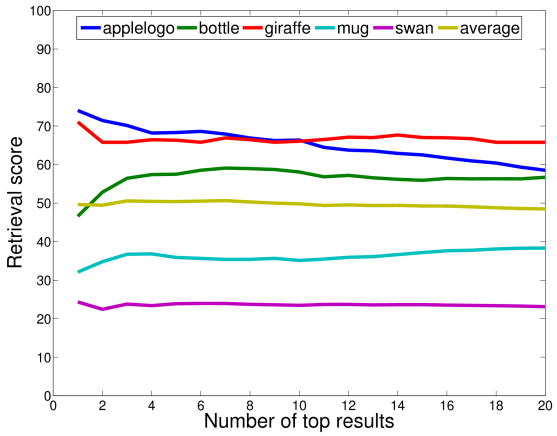

Figure 8. Recall for varying number of top ranks shows consistent retrieval results for *query-by-shape-sketch*.

However, for retrieval one is interested in the average performance over all classes, which is shown in Table 1. Here prototype models scored 58%, the average of all 700 user sketches scored 48.5% and the best single sketches scored 87%. Thus we can confirm that the sketches by Ferrari *et al*. resemble the shape prototypes quite well [8], however there are better prototypes, see second last column in Figure 6 for our best sketches. Thus on average, which reflects the typical user behavior, we can achieve a retrieval rate of 48.5%. This means using a simple hand-drawn sketch of the shape of an object, we can retrieve half of the desired images in an interactive content-based retrieval system in 50 milliseconds.

5. Conclusion

In this work we showed a novel content based image retrieval (CBIR) system, which queries a large database by means of a user-drawn sketch. This *query-by-shape-sketch* paradigm is the most intuitive extension of the current language-based semantic queries onto the visual domain. Our novel combination of shape-based features which exploit the properties of user sketches such as partial description, multiple line strokes, as well as direction and sequence of the stroke itself, and an efficient retrieval system based on hierarchical clustering and scoring allows the user to search for images by simply drawing the object of interest. This extends the current state-of-the-art by allowing an object-centered search rather than full scene retrieval.

Future work will focus on the integration of other input feature types such as color and texture, adopting a query expansion by linking *query-by-shape-*

sketch results and a *query-by-image* strategy, localization and geometric verification of sketched objects within the retrieval results and finally, investigating the universality of the shape vocabulary.

Acknowledgements We would like to thank the following people for the sketching: andrea, andy, christian, drmike, eram, george, gerhard, gerlinde, gernot, horst, joachim, julia, katl, krisi, lisa, mark, markus, martina, martinL, mat, mughero, nadja, PapaG, peterk, pmroth, robert, sabine, silke, stefan, steffi, thomas, tom, waci, werner, wm and various anonymous Open Lab Night visitors; as well as the support by the Austrian Research Promotion Agency (FFG) project FIT-IT CityFit (815971/14472-GLE/ROD).

References

[1] I. Bartolini, P. Ciaccia, and M. Patella. Query Processing Issues in Region-Based Image Databases. In *Knowledge and Information Systems (KAIS)*, 2010.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2002.

[3] A. Berman and L. Shapiro. A Flexible Image Database System for Content-Based Retrieval. *Computer Vision and Image Understanding (CVIU)*, 1999.

[4] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1986.

[5] O. Chum, M. Perdoch, and J. Matas. Geometric min-hashing: Finding a (thick) needle in a haystack. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[6] T. Cootes, D. Cooper, C. Taylor, and J. Graham. Trainable method of parametric shape description. *Journal of Image Vision Computing (JIVC)*, 1992.

[7] R. Datta, D. Joshi, J. Li, and J. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 2008.

[8] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. In *Intern. Journal of Computer Vision (IJCV)*, 2009.

[9] V. Ferrari, T. Tuytelaars, and L. Van Gool. Object detection by contour segment networks. In *Proc. European Conference on Computer Vision (ECCV)*, 2006.

[10] J. Friedman, J. Bentley, and R. Finkel. An algorithm for finding best matches in logarithmic expected time. In *ACM Transactions on Mathematical Software*, 1977.

[11] E. Gelasca, J. De Guzman, S. Gauglitz, P. Ghosh, J. Xu, E. Moxley, A. Rahimi, Z. Bi, and B. Manjunath. Cortina: Searching a 10 million + images database. In *Proc. of Conference on Very Large Data Bases (VLDB)*, 2007.

[12] Q. Iqbal and J. Aggarwal. CIRES: A System for Content-based Retrieval in Digital Image Libraries. In *Proc. of Intern. Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2002.

[13] C. Jacobs, A. Finkelstein, and D. Salesin. Fast Multiresolution Image Querying. In *Proc. of the Intern. Conference on Computer graphics and interactive techniques (SIGGRAPH)*, 1995.

[14] H. Jegou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *Intern. Journal of Computer Vision (IJCV)*, 2010.

[15] P. Kontschieder, M. Donoser, and H. Bischof. Beyond pairwise shape similarity analysis. In *Proc. Asian Conference on Computer Vision (ACCV)*, 2009.

[16] P. Kovesi. MATLAB and Octave Functions for Computer Vision and Image Processing. School of Computer Science & Software Engineering, The University of Western Australia.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. *Intern. Journal of Computer Vision (IJCV)*, 2004.

[18] A. Mikulik, M. Perdoch, O. Chum, and J. Matas. Learning a fine vocabulary. In *Proc. European Conference on Computer Vision (ECCV)*, 2010.

[19] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[20] N. Payet and S. Todorovic. From a set of shapes to object discovery. In *Proc. European Conference on Computer Vision (ECCV)*, 2010.

[21] H. Riemenschneider, M. Donoser, and H. Bischof. Using Partial Edge Contour Matches for Efficient Object Category Localization. In *Proc. European Conference on Computer Vision (ECCV)*, 2010.

[22] T. Sikora. The MPEG-7 Visual standard for conet description - An Overview. In *IEEE Trans. on Circuits and Systems for Video Technology*, 2001.

[23] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. IEEE Intern. Conference on Computer Vision (ICCV)*, 2003.

[24] H. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *Computational Science and Engineering*, 1997.