Towards a fair benchmark for image browsers

Wolfgang Müller, Stéphane Marchand–Maillet, Henning Müller, Thierry Pun

Computer Vision Group, CUI, University of Geneva
24, rue du Général Dufour, CH-1205 Geneva, Switzerland

ABSTRACT

The recent literature has shown that the principal difficulty in multimedia retrieval is the bridging of the "semantic gap" between the user's wishes and his ability to fomulate queries. This insight has spawned two main directions of research: Query By Example (QBE) with relevance feedback (i.e. learning to improve the result of a previsously formulated query) and the research in query formulation techniques, like browsing or query by sketch. Browsing techniques try to help the user in finding his target image, or an image which is sufficiently close to the desired result that it can be used in a subsequent QBE query.

From the feature space viewpoint, each browsing system tries to permit the user to move consciously in feature space and eventually reach the target image. How to provide this functionality to the user is presently an open question. In fact even obtaining objective performance evaluation and comparison of these browsing paradigms is difficult.

We distinguish between deterministic browsers, which try to optimise the possibility for the user to learn how the system behaves, and stochastic browsers based on more sophisticated Monte-Carlo algorithms thus sacrificing reproducibility to a better performance. Presently, these two browsing paradigms are practically incomparable, except by large scale user studies. This makes it infeasible for research groups to evaluate incremental improvement of browsing schemes. Moreover, automated benchmarks in the current literature simulate a user by a model derived directly from the distance measures used within the tested systems. Such a circular reference cannot provide a serious alternative to real user tests.

In this paper, we present an automatic benchmark which uses user-annotated collections for simulating the semantic gap, thus providing a means for automatic evaluation and comparison of the different browsing paradigms. We use a very precise annotation of few words together with a thesaurus to provide sufficiently smooth behaviour of the annotation-based user model. We discuss the design and evaluation of this annotation as well as the implementation of the benchmark in an MRML-compliant script with pluggable modules which allow testing of new interaction schemes (Multimedia Retrieval Markup Language).

Keywords: image browser, benchmark, structured annotation, MPEG-7

1. INTRODUCTION

Content-Based Image Retrieval Systems (CBIRS) are designed to help their user in finding images, making use of the content of each image in the collection, as opposed to labels attached to the images. The existence of large, yet un-annotated image collections, as well as the inherent limitations of image annotation motivate the research in this area.

Most current CBIRSs provide Query By Example (QBE). Here the user gives one ore more positive and negative example images in order to describe the images he or she would like to retrieve using the CBIRS. The system then will present a list of images to the user who usually has the possibility to refine his or her query, by giving additional examples from the response set. Techniques for the evaluation of such systems are close to those used in text retrieval. An overview of such techniques, adapted to the case of image retrieval is given in.[1]

While QBE addresses the question how to find images similar to a given, small set of images, interactive browsing addresses the problem of finding a given image in a collection. *I.e.* QBE addresses the problem of closely exploring a given point (or a small region) in the collection, whereas browsing systems address the problem of *mobility* within the collection.

mailto:Wolfgang.Mueller@cui.unige.ch

There are two main directions of research in image browsing: *deterministic* and *stochastic* browsing systems. Both of them present the user with successively refined overviews of the collection. The user then can express his or her preferences by marking one (optionally more, depending on the system) images as relevant or irrelevant with respect to the goal of his or her search. This information is then processed, leading to a new, refined overview of the collection.

The differences between deterministic and stochastic systems lie in the way the overviews are provided, and in the way one can navigate through the image collection. Deterministic systems provide a hierarchy which guides the image search performed by the user. The hierarchy usually is pre–calculated. This drawback is at the same time an advantage: each image search will start with the same initial selection. The browsing process could be compared to moving through a city without a map. The user has the possibility to move through the collection using fixed paths. He or she has the possibility to memorize which images will lead to which other images during the search.

In contrast to this, stochastic systems provide overviews in function of user–feedback. In contrast to hierarchical systems one has the possibility to mark multiple images as more or less relevant to the query. As a consequence, at each stage of the retrieval process, the user has so many possibilities for feedback, that a pre–calculation of the possibilities is infeasible. Thus the task varies with each image search, and it is too complex to attempt a brute–force calculation, leading to the use of Monte–Carlo methods. Monte–Carlo methods imply reproducibility *on average*, as opposed to exact reproducibility.

Both kinds of browsing systems have in common that a true test of their performance requires interaction with a user: the test user is presented with a target image, which he tries to find using the system. The performance is measured in terms of numbers of images the user had to look at. Images encountered twice are counted twice.

To our knowledge only one deep test of this kind has been done, the test of `PicHunter`.[2,3] In fact, for research groups of small size and low financial resources, tests like the one of `PicHunter` are difficult to conduct. Test users are hard to get and it is difficult to evaluate the influence of the test user's background to the experiment (*e.g.* computer vision researchers make systems look better than other office workers, but by how much?). Moreover, in deterministic systems, users cannot be used twice for the same test because they are likely to remember useful details of the last test run.

This leads to the thought of automatic benchmarking using low–level features (*i.e.* color and texture). The user is replaced by a piece of software which tries to find the target image. Cox *et al.* used this in `PicHunter` as a proof of concept backed up by real–user experiments. Vendrig *et al.*[4] used this as the only benchmarking method for their deterministic browsing system. However, in both cases the benchmark uses the same user–model as the system to be tested, thus using the testing hypothesis ("we have a useful feature set coupled with a useful learning method") for its own verification. Examples which illustrate the shortcomings of this approach are given below in § 3.2. It is argued that low–level feature based systems are not apt to function as user simulators for benchmarking other low–level feature based systems.

In this paper, we advocate a browser benchmark which is based on structured annotation. The annotation is used to simulate the learning problem a browser is facing: closing the semantic gap between visual low–level features and the semantic concepts the user is looking for. The problem of finding a good annotation method for our purpose is non–trivial. In this paper we describe the development of a structured annotation method coupled with an appropriate retrieval method for graphs with weighted edges and nodes.

As mentioned earlier, the interaction concepts of hierarchical browsers and stochastic browsers differ. In hierarchical browsers, the user needs to backtrack actively, if he or she reaches one leaf of the hierarchy. A stochastic browser will present suggestion after suggestion until the target image is found. We present a pluggable software architecture using the communication protocol MRML[5] which copes with this problem.

This paper is organized as follows: in section 2 we give a definition of the goals of an image browser. We also give examples which illustrate the problem faced by the designer both of an image browser and of the benchmark. Section 3 describes the annotation method, as well as the associated retrieval method we derived for the benchmark. This retrieval method permits QBE (as opposed to hand–formulated querying) on annotation. We describe how this method can be extended to the case of combination of annotation (*i.e.* semantic features) and low–level features. The usefulness of this method for QBE is evaluated using precision–recall graphs on 8 example queries.

Finally, this benchmark is used on a simple `PicHunter` clone (section 4.2).

2. DEFINING THE GOAL OF IMAGE BROWSERS

For defining a useful performance measure, one needs to first define what is optimal performance. Otherwise all measurements will be useless. In this paper, where we define a performance measure which is based on the simulation of user behavior, we also have to define which aspects of user behavior we want to simulate. To put it differently: *in which aspects is the system supposed to help the user?*

2.1. CBIRS using low level features

Content based image retrieval was invented as an enhancement to image annotation, and as an answer to the lack of annotation in common image collections. As the classical computer vision problem ("tell me, what's on this image") remains unsolved, CBIRS make do with low level features, sometimes accompanied by optional annotation and sophisticated interaction techniques. Using annotation in images has been shown to improve retrieval performance of low–level feature based systems.[3] This is unsurprising, because annotation contains the semantics we are not fully able to capture in low level features. However, the main issue for measuring the success of CBIRS research is *evaluating the contribution of low–level feature based systems to retrieval success.* As a consequence, we constrain the formulation of a benchmark for browsing systems on systems which do not use annotation for the search (at least while benchmarking).

2.2. Formal description of the browsing problem

In the following we assume that the user browses a given collection of images (of size N) to find one target image T. Derived problems (find one out of n images in a collection of size N) are usually easier, but not in any fundamental way. The user applies some kind of distance measure $d_{\text{semantic}}(I_1, I_2)$ between images I_1, I_2 which is mainly semantic–based, and generally a function $C \times C \to [0, 1]$ which does not satisfy the triangle inequality (*i.e.* it is not necessarily a metric). The browser, however, will apply a different distance measure, based on low level features $d_b(I_1, I_2)$. The discrepancy between these measures is the consequence of the *semantic gap.* There are now two alternatives: either the browser's measure is close enough to the user's measure to permit browsing without having to traverse large parts of the collection, or the system tries to learn from the user's feedback a measurement $d^F_{\text{browser}}(I_1, I_2)$ which approaches d_{semantic} in a sufficient manner.

2.3. Requirements for a good browser benchmark

As stated, the main use of an image browser is helping the user to close the semantic gap between low–level visual features and high–level semantics in order to browse through a collection in a way he or she understands. This is we identify as the principal requirement which should be evaluated by an image browser benchmark.

As a consequence, *any* automatic benchmark which uses low–level features only is useless for true evaluation. The process of learning a mapping between two different colorspaces is much easier than learning uniquely from user feedback that *e.g.* the user wants images with at least one dog on it. We feel that noise is insufficient to model the insufficiencies of the current feature–based user models (Fig. 1). Furthermore, such an evaluation masks the principal problem of image browsers: in many situations, meaningful answers are not possible without knowledge about the the low–level feature set, as illustrated in figure 2.

As a second consequence, if we want to get an evaluation of the discussed properties of the browser, the browser is not allowed to use annotation. Using annotation would help bridging the semantic gap in a straightforward way. It would rather work around the insufficience of low–level features for retrieval, than perform learning to improve their usefulness.

Text annotation and low–level features also are separated by a semantic gap. This semantic gap might not be as large as the semantic gap between the user's wishes and low–level features. However, it is considerable and *similar in nature* to the semantic gap a user experiences. As a consequence we suggest to benchmark image browsers by testing their target testing[2] performance when simulating users using a textual distance measure d_{text}. The distance $d_{\text{text}}(I_1, I_2)$ between two images is determined using text retrieval techniques on the annotation of I_1 and I_2. Details are described below.

Figure 1. *An example where the semantically more distant image is considered closer to the query:* Viper[6] *(in high-speed-low-quality mode) considered the middle image closer to the left image, than the right image. This is due to matching the black trousers of the man in the middle picture to the dog jumping in the left picture. However, clearly the semantics of the right image is closer than the left.*

Figure 2. *An example that illustrates that in many cases, a sensible answer does not exist. During the browsing process, often the user is confronted with questions like the following:* "What is more similar to the image of the man and the sitting dog: the mountain or the pound note?" *Stochastic browsers provide the user a possibility to decline an answer if the selection does not offer the possibility for sensible feedback.*

3. RANKED QBE ON STRUCTURED ANNOTATION

We now focus on a semantic–based distance measure between images. At first glance, this problem seems to be easily tractable using classic text–retrieval techniques. However, this is not the case, as described in the following subsection.

We then describe the structured annotation approach we adopted, along with the retrieval method we used on the annotation. We give a performance–evaluation of this annotation, and compare it to the performance of a similar, unstructured approach. As an outlook to further work, we put this into relation with ongoing efforts in the MPEG–7 standardization process.

3.1. Differences with classical text retrieval

Textual information retrieval is an old area of strong economic interest. Much research has been done in the last 40 years. The successful establishment of a common benchmark by the Text REtrieval Conference (TREC[7]) has rendered results comparable and has created a general competition for the best text retrieval solution.

Presently, the systems performing best in TREC use very little linguistic or semantic knowledge. As E. Vorhees[8] states text "is regarded as little more than a bag of words". To summarize the basic principle of many systems: for each term (word) of the query, each document which contains this term receives a score, depending on the frequency of the term in the document *tf*, as well as the frequency of the term in the collection *idf*, leading to $tf \cdot idf$ measures. The rationale is: if a document contains a term rare in the collection, this distinguishes the document well from others. If a document contains a term frequently, it is supposed to be important for the document.

Natural language processing (NLP) techniques have had little success up to now. It has been identified as one reason[8] that the disambiguation techniques are too error–prone: the precision gained by more accurate modeling of the word relationships is lost by trusting too much in wrongly established word relationships.

However, in our case the situation is different:

- TREC deals with documents of kilobytes in size whereas annotation usually is much shorter. As a consequence, statistical measures like $tf \cdot idf$ will produce less accurate results.
- While chances are high that in well written long texts multiple synonyms of one meaning are used, usually only one synonym of a given word will appear in a very short text. Thus the analysis of short texts requires better determination of the true word sense.
- In the scenario of a query that has been formulated by hand, one can assume each query term to be relevant to the user. However, in our case we are interested in the distance between documents, *i.e.* we are interested in QBE instead of hand–formulated queries. As a consequence, not every query term is also relevant to the user who gave the example.
- Annotation is made for retrieval purposes. Adding structured annotation to an image takes only little more time than adding unstructured annotation to an image. We can use the structured annotation for replacing the faulty disambiguation step by hand–made disambiguation.

Consider the following example: A database contains an image, annotated by the caption `A dog jumping over a bar. Bushes in the background. People in the background.`. Using this as a positive example in a QBE query on captions, a normal text query might retrieve `A statue. Bushes in the background. People in the background.`. The result `A dog jumping over a bar.` would get a lower rank, as less items match with the query.

Syntax example	Meaning
`setting(athletics).`	the word `athletics` describes the image as a whole.
`actor(dog,§dog).`	the word `dog` designates an entity which is capable of action. A graph node is created for this entity. It can be referred to using the word `§dog`. The node ID `§dog` is visible for all subsequent facts that refer to the same image. *i.e.* the node name §dog will be reusable in another image.
`actor(dog).`	Abbreviation for `actor(dog,§dog).`. Similar abbreviations exist for other tags
`actorS(dog).`	Many dogs (`§dog`), an abbreviation for a combination of several tags.
`actorS(dog,10).`	10 dogs (`§dog`).
`thing(house,§house).`	The `house` designates an entity incapable of action. `§house` designates the node created for this entity.
`modifies(black,§dog).`	Modifies, further describes an entity node which has been created.
`enumerates(10,§dog).`	Specialization of `modifies` for numbers.
`action(run,§run).`	Creates a node describing the action run.
`staticAction(stand,§stand).`	For discerning verbs like sit, lie, stand.
`performs(§man,§hit).`	The entity `§man` performs the action `§hit`: a man is hitting (in the following is assumed that `§man`, `§hit` *etc.* have been instantiated in a sensible way).
`isPerformedOn(§hit,§dog).`	a dog is hit.
`using(§hit,§stick).`	a stick is used for hitting.

Table 1. *This table describes the main syntactic elements which were used for structured annotation. We omitted some instructions for spatial relationships on an image.*

Intuitively, each item we want to describe has to be described using at least one word, even if it is of little importance. Because of the shortness of the annotation text, it is a matter of chance (*i.e.* the statistics of the database), if the term we employed for the background item is rare or frequent in the database, and thus, which weight it will receive. In other words, there is a need for pre–weighting of terms. This can be derived from structured annotation, as described next.

3.2. The structure of the annotation

As it was expressed in the last subsection, structuring the annotation causes little overhead. Furthermore there is the need for structuring the annotation, in order to add information about the importance of the different items of the annotation. Structuring the annotation is meta–annotation. Its main advantage for us lies in enabling the use of *a–priori* information about the importance of items in the annotation text.

In our annotation effort, we focused on emphasizing the importance of participation in an action as opposed to passiveness. The rationale of this is that most of the time, if there is action on an image, the parts of the image that are not implicated in the action are less important, they are to be considered as part of the background. Furthermore, we wanted to be able to express subject–object relations: classic text retrieval methods will not distinguish between `Man bites dog.` and `Dog bites man.` However, while the first example surely *is* a news item, the second happens every day.

We designed a small set of relationships, not with the intention of being linguistically complete, but with a view on our image collection and the interesting relationships between items in this collection. For structuring our annotations, we implemented a small language which compiles facts with Prolog–like syntax into Prolog. The main syntactic elements of the annotation are described in Table 3.1.

This permits the expression of simple semantic networks. The annotation presented here is not a full fledged linguistic annotation. However, it captures the basic relationships between items of the annotation. Please note that the annotator has to name by himself the nodes of the semantic net for one image, but that the burden to find non-conflicting node IDs for the whole collection is taken care of by the compiler that translates this structure into Prolog. The language presented here has the advantage that while being close enough to programming languages for testing purposes, the syntax is short and simple to memorize. The left half of Fig. 3 shows the semantic network derived from the annotation

```
actor(dog).
actor(man).
action(bite).
performs(§dog,§bite).      % read: "dog performs bite"
isPerformedOn(§bite,§man). % read: "bite is performed on man"
```

3.3. The retrieval method

It is well known that graph matching is a difficult problem. Most graph problems are NP-complete. However, in our situation, we can make use of the structural knowledge of the graph, and our knowledge about what interests us in the graph to simplify the problem.

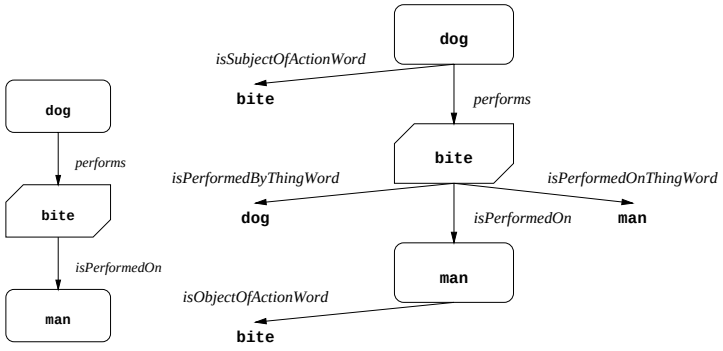


Figure 3. *A basic example of structured annotation as described in the text. Initially, the annotation is structured as shown in the image on the left. This structure is automatically augmented as shown in the figure on the right. This becomes useful in inexact matching.*

For matching two images, we take a three–step approach:

1. Identify similar nodes between query and match. Assign a score to each matching node

2. Verify the relationships between pairs of similar nodes in query and match. Increase the score for each pair of similar nodes in similar relation.

3. Sum the score for each matched node. The result is the score for an image.

Evidently, the crucial step is the first one: without proper identification of similar nodes, the verification of the relation will be impossible. However, in our case, this identification is simple. We store with each actor the verb of the action it performs, the verb of which it is object, *etc.*

This means that we perform a simple, unstructured query for the features of each node that are connected to one node only (*i.e.* `modifies`, `enumerates`, `isSubjectOfActionWord`, *etc.*) which we use as a basis for a greedy identification of nodes with similar function in the query and matching images. Afterwards we use these results for verifying the relationships between nodes. As a consequence, this retrieval method approaches classic inverted–file text retrieval algorithms in efficiency.

Weighted queries on conceptual graphs are also described in.[9,10]

3.4. Performance of the annotation

In this section we will describe the performance of the annotation when doing QBE on an annotated collection. The goal is to show, that this annotation gives a good "one-shot" retrieval performance, making the annotation suitable as the basis for the simulation of a real user in a browsing scenario. Our results are compared to the results of applying classical text retrieval methods on unstructured annotation with the same content. This unstructured annotation was derived from the structured annotation by removing the structuring elements from the structured annotation.

For this, the described annotation and retrieval scheme was used on the 500 images provided by the *Télevision Suisse Romande*, the French-speaking swiss television station. This image collection was chosen for its diversity. It contains scenes of varying complexity and varying degree of action.

The images were presented in portions of four images to the annotator on a 1024×768 pixel 13.3" LCD panel. The resolution per image was 256×256 pixels. The annotator (the first author) had the opportunity to scroll back and forth both in the annotation and in the image collection. The average time spent per image was about 5 minutes. After the complete annotation was done, a debugging pass was performed. Here, drifts in annotation strategy (detected using test queries) as well as typographical and syntax errors were corrected.

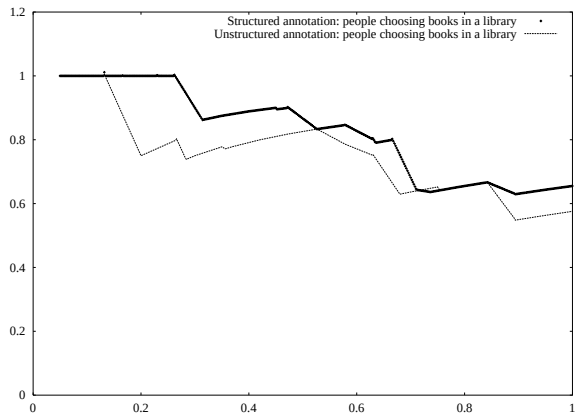
We then performed 8 QBE queries, using the annotation scheme and the retrieval method described in § 3.3. The performance of these QBE queries were evaluated using relevance data which were collected for the experiments with *Viper* .[6] User data was collected for five users. Each one performed the queries by hand, thus providing for each image a list of images relevant to the query. We kept all results for each user, thus storing the whole range of user behavior. This enabled testing the performance on relevance feedback, as shown in.[11] In our present experiment, these relevance judgments were used to obtain precision–recall graphs of one–shot–queries on structured annotation.

For all queries, the structured annotation performs at least as well as the equivalent unstructured annotation (derived from the structured annotation by suppressing the structure). However, once again, it becomes clear that the problem of QBE for images is ill–posed: what is considered as relevant differs widely between the test subjects. With most query images both structured and unstructured annotation reached perfect performance for at least one test user. With some other images there is an advantage for the structured annotation, as shown in 4.

Both annotation methods performed very badly on test images that showed buildings as the only noticeable image item. We see as an explanation that both the annotator and the test users have no architectural background. So the relevance data were rather a product of the visual impression than of the semantics. However, in architecture and art, established classification methods exist.[12] These might be included in future versions of our annotation.

We also experimented with the use of a thesaurus for improving the retrieval performance. We found that in our scenario, synonym sets and synonym disambiguation can be used in a beneficial way (*e.g.* volume–book instead of volume of liquid, for example). In using WordNet,[13] we experienced performance improvements when using WordNet synonym sets instead of the words of the annotation. We would like to underline, that also in this case, disambiguation has been done by hand in order to improve the query result. Trying to add WordNet hypernyms (*i.e.* generalizations) to each annotation item in a straightforward way, degraded the query results. However, we suggest adding hand–selected hypernyms where appropriate (*e.g.* policeman–man).

Most important in this context is that the structured annotation produced long strings of images that are *semantically consistent* with the query. As a consequence, we found that our annotation can be used as user simulation for enabling a benchmark for image browsers.




```
% % some words describing the setting
setting(books).
setting(library).
setting(inside).
% % people looking at books, and choosing
them thing(library).
modifies(public,§library).
thingS(bookshelf).
actor(person,5).
action(read).
performs(§read,§person).
action(choose).
performs(§choose,§person).
thingS(book!2).% WordNet-sense 2 of "book"
isPerformedOn(§read,§book).
isPerformedOn(§choose,§book).
action(stand).
performs(§person,§stand).
```

Figure 4. *Annotation example, describing 5 people in a library standing next to bookshelves, reading and choosing books. The precision–recall graph compares the performance of structured and the corresponding unstructured annotation for this example.*

3.5. Putting our retrieval method into context with MPEG-7

We would like to emphasize that we see our retrieval method for structured annotation as interesting for an integrated retrieval method for MPEG-7 data. MPEG-7 will standardize many types of image annotation. The challenge for MPEG-7 retrieval systems will be the fusion of all the annotation items in order to provide a common, easy–to–use interface. Our retrieval method presented here can be easily extended to include properties of still–image–segments as new nodes in the semantic network, also adding an additional type of node relation to the semantic network. The retrieval methods themselves would stay the same.

4. THE BENCHMARK

In the past sections we have first described our needs for a distance measure for images which is closer to the human way of thinking than low–level features. We then described an annotation method as well as an associated retrieval method that permits QBE on images using the annotations. Now we will proceed with the description, on how to use QBE on annotation for benchmarking browsing systems.

4.1. A benchmarking system with pluggable components

We present a benchmark that uses MRML[5] (also described in this volume), an extensible query protocol for content-based image retrieval systems. Here it serves to separate the benchmarking system and benchmarked system.

Even with the standardized communication layer, the following modules change when changing the retrieval method:

Initial query formulation: allowing for QBE, hand–formulated queries, or just requesting an initial selection of images.

Relevance feedback method: allowing the distinction between QBE and browsing queries. Determining if we are allowed to mark irrelevant images as negative examples, and if we are allowed to mark multiple images?

Determining relevance: in a QBE scenario, the only interesting parameter is the membership of a retrieved image in the set of images deemed relevant by test users. In browsing queries, *relative relevance* is important: at each step, we want to choose images which are more relevant than others.

Our benchmarking architecture allows simple exchange of these modules. The resulting program automatically performs a user simulation, storing the results of each query step in a database. These data can then be evaluated using programs that query the database.

4.2. Benchmarking an imitation of PicHunter

This benchmarking system (BS in the following) was applied on a system using PicHunter's Bayesian retrieval method. Our PicHunter-like system used color histogram distance and an image shape spectrum[14] based distance measure. In the following, we will call this system QuickHunter.

For performing the benchmark we need three entities: The *benchmarking system* (BS), that dispatches queries to the *benchmarked system* (QuickHunter in this case) and an *annotation-based query engine* (AQE, as described in § 3.3), which provides the distance measure.

For the benchmark, the BS performed a so-called *target test* for a list of images. For each target image, the BS performed a query by example on the AQE. The query result was pruned by hand so that it conained images with a relation to the query only. Normalized, this ranked list served as d_{text}. Now the BS requested a (random) selection of 9 images from QuickHunter. The image of the selection with the smallest d_{text} was marked positive, the one with the biggest d_{text} was marked negative, and this query was submitted to QuickHunter. QuickHunter used this to calculate a new selection of 9 images, always based on its d_{browser} distance measure. On finding the query image, the search was re-initialised, and the process was repeated for the next target image.

For each target image, we counted the number of images which were shown to the user before the target was found by QuickHunter. As query images, we used the same images as for the evaluation of the annotation. The results are shown in Table 2. On average, the QuickHunter needed to scan 170 images before the target was found, being more efficient than random search (250 images) by $\approx 30\%$. We also remark that, in these experiments, the performances of QuickHunter varied by a factor of ten, depending on the "difficulty" of the target. As a consequence, we do not think that at the current state of research, the target testing performance of a system can be summarized by a single number.

Obviously, the combination of our image collection and our benchmarking method is very hard for systems that do not try to adapt their user model during retrieval. A further difficulty is that, in realistic collections, the number of images that are in any relationship with the target image is very small. This limits the possibilities of the BS to give useful feedback, like it does with real users.

It is not surprising, that at the current state of CBIRS research this benchmark is very hard for the benchmarked system. Therefore we propose a migration path towards semantic benchmarking: use a superposition of a semantics–derived and a visual (*i.e.* low-level feature) distance measure for user simulation. The weight with which both distance measures are superimposed could then be used to describe the degree of difficulty of the benchmark.

Query	#images	remarks
One dollar	23	perfect user agreement to the annotation,
500 German marks	50	many almost–relevant images
Corner view of building	140	medium user agreement, many almost–relevant images
Library	150	high user agreement visually inhomogene relevant set
Parliament	165	dto.
Lemons	241	perfect user agreement to the annotation, small set of related images (3)
Harbour	255	perf. agreement to annotation, but very small set of vis. similar rel. images
Russian palace	315	very bad user agreement with this annotation

Table 2. *Benchmarking results for a PicHunter like system, QuickHunter, for 8 queries on TSR500: #images designates the number of images that had to be seen by the simulated user before finding the target.*

5. CONCLUSION

We proposed an automatic semantic–based benchmark for image browsing systems. The advantage of such a benchmark is that it is memoryless, and that it is not influenced by imponderables like the previous experience of test users.

We based our automatic benchmark on structured annotation that is augmented using a thesaurus. For this benchmark, we developed a fast query method for semantic networks, that performs ranked similarity queries using inverted files, followed by a stage where more elaborate matching is performed.

We see the use of this work as two–fold: First, in the near future, we will study the interweaving between annotation and still image features, especially still image segments. Trivially, the retrieval method proposed allows for inclusion of derived visual features with and without annotation. Our studies regarding this topic will be described in a later publication. Secondly, the benchmark devised in this paper constitutes a tool, both for the development and for the evaluation of image browsing systems. We hope it will trigger more research for intelligent systems that learn the *semantics* of a query during the querying process.

ACKNOWLEDGMENTS

The authors wish to thank *David McG. Squire* and *Audrey M. Tam* for many fruitful discussions. This project is supported by the Swiss National Foundation for Scientific Research under grant number 2000-052426.97.

REFERENCES

1. H. Müller, W. Müller, D. M. Squire, S. Marchand-Maillet, and T. Pun, "Performance evaluation in content-based image retrieval: Overview and proposals," *Pattern Recognition Letters* , 2000.
2. I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos, "Target testing and the `PicHunter` Bayesian multimedia retrieval system," in *Advances in Digital Libraries (ADL'96)*, pp. 66–75, (Library of Congress, Washington, D. C.), May 13–15 1996.
3. T. V. Papathomas, T. E. Conway, I. J. Cox, J. Ghosn, M. L. Miller, T. P. Minka, and P. N. Yianilos, "Psychophysical studies of the performance of an image database retrieval system," in *Human Vision and Electronic Imaging III*, B. E. Rogowitz and T. N. Pappas, eds., vol. 3299 of *SPIE Proceedings*, pp. 591–602, July 1998.
4. J. Vendrig, M. Worring, and A. W. M. Smeulders, "Filter image browsing: Exploiting interaction in image retrieval," in Huijsmans and Smeulders,[15] pp. 147–154.
5. W. Müller, H. Müller, S. Marchand-Maillet, T. Pun, D. M. Squire, Z. Pečenović, C. Giess, and A. P. de Vries, "MRML: A Communication Protocol for Interoperability and Benchmarking of Multimedia Information Retrieval Systems," in *Internet Multimedia Management Systems*, N.N., ed., vol. 4210 of *SPIE Proceedings*, (Boston, Massachusetts, USA), November 6–8 2000. (SPIE Information Technologies 2000).
6. D. M. Squire, W. Müller, and H. Müller, "Relevance feedback and term weighting schemes for content-based image retrieval," in Huijsmans and Smeulders,[15] pp. 549–556.
7. "Text REtrieval Conference (TREC)." `http://trec.nist.gov/`, 1999.
8. E. Voorhees, "Natural language processing and information retrieval," in *Information Extraction: Towards Scalable, Adaptable Systems*, M. T. Pazienza, ed., vol. 1714, pp. 32–48, Springer Verlag, 1999.
9. I. Ounis, "A Flexible Weighting Scheme for Multimedia Documents," in *Proceedings of the 10th DEXA International Conference on Database and EXpert Systems Applications*, pp. 392–405, (Florence, Italy), August 1999.
10. I. Ounis and M. Pasca, "Modeling, indexing and retrieving images using conceptual graphs," in *Proceedings of the 9th DEXA International Conference on Database and EXpert Systems Applications*, pp. 226–239, (Vienna, Austria), August 1998.
11. H. Müller, W. Müller, D. M. Squire, S. Marchand-Maillet, and T. Pun, "Strategies for positive and negative relevance feedback in image retrieval," in *Proceedings of the 15th International Conference on Pattern Recognition (ICPR 2000)*, IEEE, (Barcelona, Spain), September 2000.
12. J. Greenberg, "Intellectual control of visual archives: A comparison between the art & architecture thesaurus and the library of congress thesaurus for graphic materials," *Cataloging and Classification Quarterly* **16**(1), pp. 85–117, 1993.
13. C. Fellbaum, ed., *WordNet - An Electronic Lexical Database*, MIT Press, Cambridge, 1998.
14. C. Nastar, "The image shape spectrum for image retrieval," Tech. Rep. RR-3206, INRIA, Rocquencourt, France, July 1997.
15. D. P. Huijsmans and A. W. M. Smeulders, eds., *Third International Conference On Visual Information Systems (VISUAL'99)*, no. 1614 in Lecture Notes in Computer Science, (Amsterdam, The Netherlands), Springer-Verlag, June 2–4 1999.