Semantic Feature Layers in Content-based Image Retrieval: Implementation of Human World Features

Horst Eidenberger and Christian Breiteneder

Vienna University of Technology, Institute of Software Technology and Interactive Systems,
Favoritenstrasse 9-11 – 188/2, A-1040 Vienna, Austria
{eidenberger, breiteneder}@ims.tuwien.ac.at

ABSTRACT

The major problem of most CBIR approaches is bad quality in terms of recall and precision. As a major reason for this, the semantic gap between high-level concepts and low-level features has been identified. In this paper we describe an approach to reduce the impact of the semantic gap by deriving high-level (semantic) from low-level features and using these features to improve the quality of CBIR queries. This concept is implemented for a high-level feature class that describes human world properties and evaluated in 300 queries. Results show that using those high-level features improves the quality of result sets by balancing recall and precision.

1 INTRODUCTION

Content-based Image Retrieval (CBIR, [2]) approaches aim at finding images that are *semantically similar* to a given query (often a single example image). In this definition, 'semantically similar' is meant in the sense of human visual similarity perception (in CBIR publications mostly just called 'high-level'). The methods used to satisfy this demand are based on numerical feature extraction (e.g. with signal processing and computer vision techniques) and (metric-based) distance measurement. This approach is usually referred to as 'low-level'. Now the problem of most (general-purpose) CBIR approaches is bad quality in terms of recall and precision. As a major reason for this, the semantic gap has been identified ([9]). This is the gap between the high-level requirements of CBIR and the low-level implementation.

In this paper we describe a novel approach to reduce the impact of this semantic gap. Usually, iterative refinement by relevance feedback is used to minimize the semantic gap in CBIR systems ([7], [12]). We follow a different path by deriving high-level (semantic) from low-level features and using these features to improve the quality of CBIR queries. We show by an example prototype implementation and evaluation the idea of the approach.

The results of this paper are part of the Visual Information Retrieval project VizIR. The VizIR project aims at the following major goals:

- Implementation of a modern, open class framework for content-based retrieval of visual information as basis for further research on successful methods for automated information extraction from visual media, definition of similarity measures and new, better concepts for the user interface aspect of visual information retrieval.
- Implementation of a working prototype system that is fully based on the visual part of the MPEG-7 standard. Obtaining this goal requires seeking for suitable extensions and supplementations of the MPEG-7 standard.
- Development of integrated, general-purpose user interfaces for visual information retrieval.
- Support of methods for distributed querying, storage and replication of visual information and features and methods for query acceleration.

To achieve these goals state-of-the-art software development is necessary. VizIR is based on reverse engineering and the Rational Unified Process ([6]). The output of VizIR will be available to the public. The overall goal of VizIR is providing the research community with a flexible tool for experiments. See [3] for more information on VizIR.

The rest of the paper is organized as follows. Section 2 points out relevant related work, Section 3 describes the idea of semantic features, in Section 4 we outline the design of the Human World Feature class (HWF), Section 5 describes the implementation of HWF in our test environment, Section 6 discusses experimental results and finally, Section 7 sketches our next activities in the context of this paper.

2 RELATED WORK

Subsequently, we will review the semantic gap problem, point out a second current approach for semantic feature extraction and briefly describe the descriptor definition language (DDL) of the MPEG-7 standard, that will be used to describe HWF.

According to [9], the semantic gap can be defined as the space of disappointment between the high-level intentions of CBIR and the low-level features that are used for querying. The size of the gap in current general-purpose systems ranges from 60% to 80% of querying performance (recall and precision, e.g. in [10]). In his keynote speech at the Visual Information Systems conference 2002, William Grosky described a semantic feature extraction method related to the Semantic Web project ([8]) that should help to reduce the semantic gap. Basically, the idea is to integrate close distant information into the feature extraction process. For example, on a webpage, image features are not just derived for the area of each image but for an area that includes the image and the text around it. Thus, semantic information is integrated in the feature vectors. The

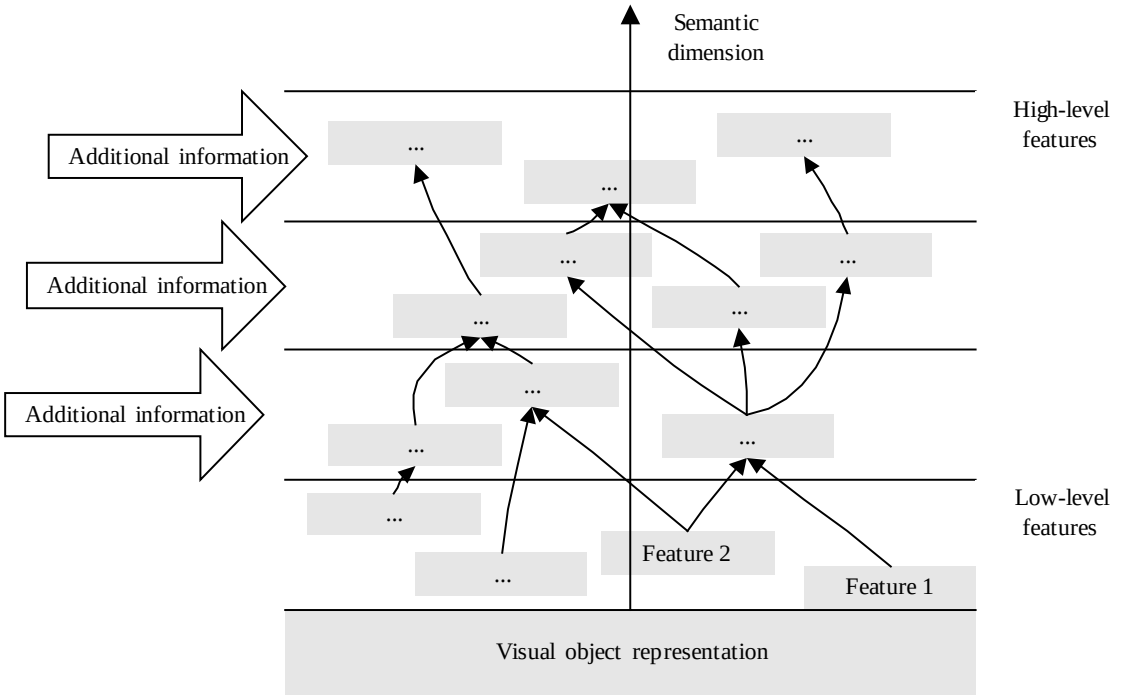

Figure 1. Semantic feature layers. Features on higher levels are based on the Descriptors of features on lower levels. Together with additional information they derive new Descriptors on higher semantic levels. Additional information includes modeling information, statistical information and domain knowledge. This model should help to narrow the semantic gap.

problem of this method – from our point of view – is that it is difficult to argue, *why* adding image rendered text information to the image feature extraction process should improve the quality of retrieval results.

The semantic features introduced below are defined on the basis of the MPEG-7 Descriptor Definition Language (DDL). MPEG-7 defines Descriptors (D), Descriptor Schemes (DS) and the DDL. DS are containers of D and DS. The DDL is a uniform method for the description of D and DS. Essentially, the DDL is the XML Schema Language, extended by a few custom data types (like matrices, histograms, etc.). As the authors of [5] state, 'the DDL is not a modeling language such as Unified Modeling Language (UML, [11]) but a schema language to represent the *results* of modeling audiovisual data.'. Thus it is impossible to model the usage of additional knowledge in D and DS.

3 SEMANTIC FEATURE LAYERS

The idea of semantic feature layers (SFL) is the design of semantically related feature classes that are based on features of lower levels and include additional knowledge (see Figure 1). Additional knowledge can be comprised of modeling information, domain knowledge, statistical information, etc. and be expressed as data (e.g. a color covariance matrix) or as algorithms (e.g. a sophisticated distance measurement algorithm). SFL should help to reduce the size of the semantic gap.

SFL are more than DS. DS define hierarchical relationships of static Descriptors and other DS. In SFL, Descriptors do not remain static on higher levels but are transformed by additional knowledge to more specific (semantic) representations. Using SFL in addition or instead of low-level features has two major advantages:

1. It is possible – in the context of the SFL – to perform high-level queries without the need to translate them to queries on low-level features. This should lead to better results.
2. Queries are much faster, because of simpler feature vectors and simpler querying methods. The integration of additional knowledge on the basis of low-level features will in most cases lead to a compression of the high-level feature vectors. This process is performed offline during the feature extraction process. Querying methods can be simpler because no mapping is necessary and feature vectors are simpler.

SFL are an abstraction of low-level features. In the next section we will introduce an example of a SFL for the description of human world properties in images.

4 HUMAN WORLD FEATURES

The world of visual objects (from the human point of view) can be split into two groups: nature-originated objects (e.g. landscapes, trees, etc.) and human-originated objects (e.g. equipment, houses, etc.). The idea of the human world properties SFL (HWL) is the definition of features that describe typical properties of human-originated objects and scenes. This is useful, because most images consist of both types of objects and the relationship of them is often typical for a certain image group (cluster, application domain, etc.). For

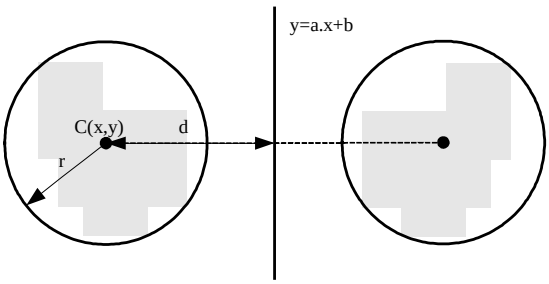

Figure 2. Semantic symmetry feature: symmetry axis detection. The symmetry axis for two objects is derived from the circles around them.

example, with HWL features images of family photos at Christmas can easily be distinguished from family photos in summer (at least in the colder regions of the world), because Christmas photos are usually made indoors (with a lot of human-originated objects in the background) while summer photos are usually made outdoors (with significantly more nature-originated objects in the background).

We have identified three major properties of human-originated objects that can be relatively easy described with numerical feature vectors:

1. Geometry. Humans love to create objects with the major properties of Euclidean geometry: straight lines and right angles. These properties are hardly present in natural objects.
2. Harmony. This includes human characteristics like the harmonic application of colors (matching colors and color shades), harmonic textures and the regular arrangement of objects in scenes. Even though the human preference for harmony is presumably originated in natural characterization it furthermore has a cultural component that makes it different from the harmony appearing in natural scenes.
3. Symmetry. This does not refer to the mathematical symmetry term (concerning symmetric objects, this symmetry exists in nature as well) but to the *symmetric arrangement* of objects (represented by more or less coarse object representations) that can be symmetric (e.g. a row of windows), mirrored (e.g. semidetached houses) or repetitive (e.g. a row of computers).

These properties are employed to judge whether an object appearing in a scene is human-originated. They can be represented by feature classes that can be based on arbitrary low-level features that include spatial or geometric information (e.g. localized color histograms, object descriptions, edge histograms, etc.). As an example, let us detail the algorithm for an implementation of the symmetry feature. It consists of three steps (see Figure 2):

1. Extraction of all occurrences of the underlying feature in the visual object. The underlying feature can be every feature not invariant against mirroring and that may be contained multiple times in a visual object (e.g. spatial color distribution, texture moments, etc.).
2. Extraction of all mirrored occurrences of the underlying feature in the visual object. Each found object is represented by the radius and center of the circle around it.
3. Detection of the parameters of the symmetry axis for found pairs.

The Descriptor of this symmetry feature (according to a specific underlying feature) could be the following vector:

$$(C(x,y),r,a,b,d)^t$$

where *C(x,y)* and *r* are defined as above (for a not mirrored object), *a* and *b* are the parameters of the symmetry axis and *d* is half of the shortest distance from *C(x,y)* to the symmetry axis. For our tests we used an even simpler implementation of the symmetry feature. The next section is dedicated to this matter.

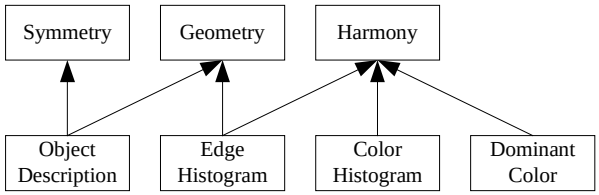

Figure 3. Human world feature layer implementation. The three high-level features are based on four low-level features.

5 IMPLEMENTATION

For experimental evaluation (see Section 6 for results) we have implemented a simple version of the HWL. It consists of three features, one for each of the properties above. These features are based on four low-level features. Figure 3 shows the dependencies.

The first low-level feature derives a simple object description that includes the object size (in macroblocks), the circularity of the border (as defined in [4]) and the position in the image for the first five objects. A macroblock has one 64th of the width and height of the image. The edge histogram has four bins for all edges in an image with length smaller than one macroblock, one to two macroblocks, two to four macroblocks and more than four macroblocks. Additionally, we use a global color histogram with nine bins and the MPEG-7 dominant color feature with two bins. The first three low-level features use Euclidean distance functions for dissimilarity measurement. The dominant color feature uses the following function to compare two objects $A=(c1_A,c2_A)$ and B.

$$d(A,B) = 0.5u(c1_A,c1_B) + 0.2u(c1_A,c2_B) + 0.2u(c2_A,c1_B) + 0.1u(c2_A,c2_B)$$

where *u(x,y)* is defined as follows:

$$u(x,y) = \begin{cases} 0 ... if\ x = y \\ 1 ... otherwise \end{cases}$$

The weights were set based on heuristics. The output of

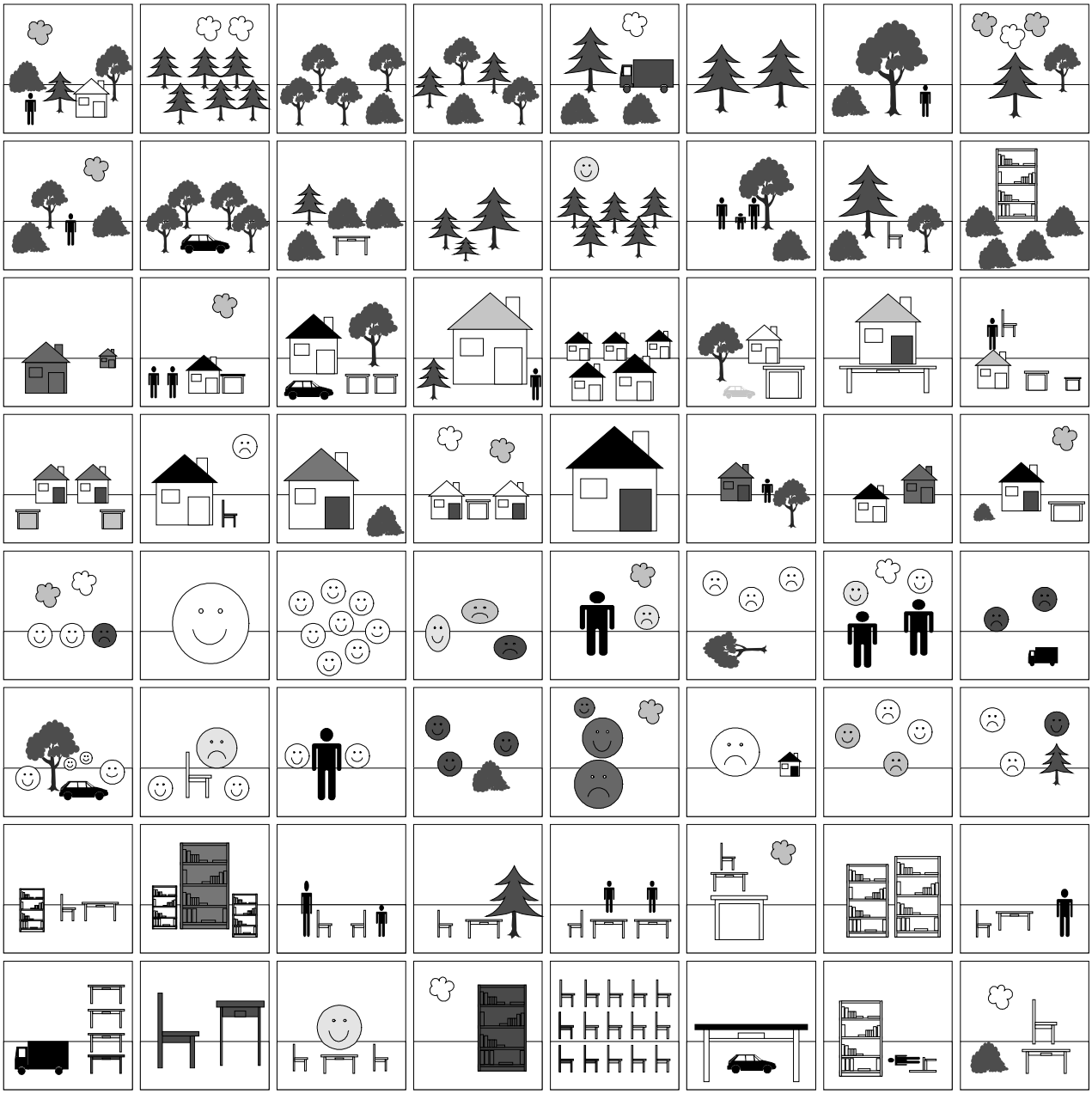

Figure 4. Test images and ground truth. The collection consists of four groups with 16 images each. The four groups are: images of forests (first and second row), images of houses (third and fourth row), images of faces (fifth and sixth row) and images of equipment (seventh and eighth row).

all distance measures is normalized to the interval *[0,1]*.

The geometry feature is based on the object description and the edge histogram. It measures the number of straight lines with significant length (longer than two macroblocks; derived from the edge histogram) and the number of right angles in an image (derived from the circularity values). We define the following MPEG-7 descriptor:

```
<complexType name="GeometryFeature">
   <element name="StraightLines"
    type="unsignedInt" use="required"/>
   <element name="RightAngles"
    type="unsignedInt" use="required"/>
</complexType>
```

The distance of two descriptors $A=(sl_A,ra_A)$ and B is measured with the following distance function.

$$d(A,B)=\sqrt{\frac{(sl_A-sl_B)^2+(ra_A-ra_B)^2}{2}}$$

This is basically an Euclidean distance.

The harmony feature is based on the edge length histogram, the color histogram and the dominant color feature. It has three bins for the amount of activity in an image, the number of color gradations and the color type (warm, cold, grey-scale). The activity in an image is measured as the variance of edge lengths. The MPEG-7 descriptor for the harmony feature is defined as follows.

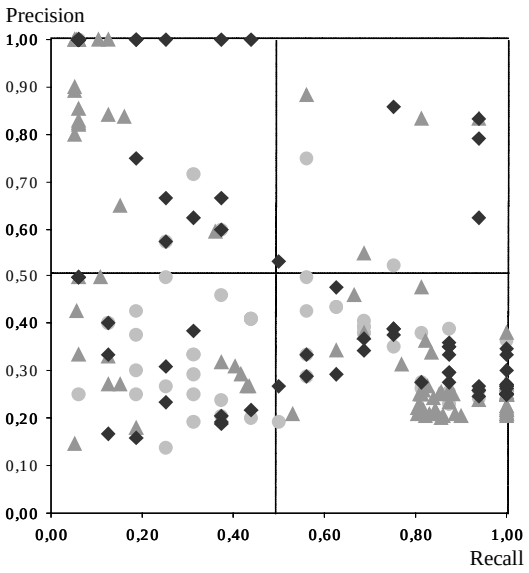

Figure 5. Experimental results for 300 queries. Triangles represent the query results for the low-level features, circles the results for the HWF and rhombs the results for queries on all feature classes.

```
<complexType name="HarmonyFeature">
   <element name="Activity"
    type="unsignedInt" use="required"/>
   <element name="ColorShades"
    type="unsignedInt" use="required"/>
   <element name="ColorType"
    type="unsignedByte" use="required"/>
</complexType>
```

The distance of two descriptors $A=(act_A,cg_A,ct_A)$ and B is measured with the following distance function.

$$d(A,B)=\sqrt{\frac{(act_A-act_B)^2+(cg_A-cg_B)^2+u(ct_A,ct_B)}{3}}$$

where *u()* is defined as for the distance function of the dominant color feature.

The symmetry feature is based on the object description feature. It counts the number of symmetric objects (equal descriptions with equal size) and the number of repeated objects (equal descriptions with different size). We define the following Descriptor:

```
<complexType name="SymmetryFeature">
   <element name="Symmetries"
    type="unsignedInt" use="required"/>
   <element name="Repetitions"
    type="unsignedInt" use="required"/>
</complexType>
```

The distance of two descriptors *A* and *B* is measured with the same function as for the geometry feature.

All features (low-level and HWF) and a querying engine that is based on our Query Model concept ([1]) were implemented as Perl objects in our test environment. Perl was chosen because it allows rapid prototyping. The next section explains how we tested the HWL features and the results we got.

6 EXPERIMENTAL RESULTS

All experiments were done on a collection of 64 synthetic images. This collection consists of four groups with 16 similar images each. Figure 4 depicts the test database. Each group consists of two rows. The four groups (ground truth) are: images of forests (first and second row), images of houses (third and fourth row), images of faces (fifth and sixth row) and images of equipment (seventh and eighth row). Each image was constructed from a stencil with 14 basic icons in Microsoft Visio (the image collection and the icon stencil can be obtained from the authors). We chose this image collection because it is – although the images are synthetic – a hard test for the SFL concept and the HWL implementation. It is a hard test because these images do not contain much information and it is difficult to derive more information with high-level features than the powerful low-level features (color histogram, object description) already do.

The hypothesis of our experiments was that *using SFL reduces the impact of the semantic gap*. This was tested in the following way:

- The HWF defined above were used as an example of an SFL. We did 300 valid queries: 100 with the low-level features, 100 with the HWF features and 100 with all features. The parameters of these queries were selected automatically (query example, threshold parameters, see [1]).
- A query was defined as valid, if the result set was not empty. This was the only restriction in the automatic evaluation process.
- The reduction of the semantic gap was measured by the change in the quality of result sets. Quality was measured with recall and precision. The ground truth from above was used for evaluation.

Querying was done by selecting an example image from the given collection and setting threshold values for the used features. The thresholds are upper limits for the distance from an image to the query example. If an image exceeds the threshold for a certain feature, it is discarded from the querying process. The result set contains only the images with a distance (for every feature) to the query example that is not greater than the feature-specific threshold.

Figure 5 shows the results of all queries. Triangles represent the query results for the low-level features, circles the results for the HWF and rhombs the results for all features. We have split the diagram in four areas: *excellent* (recall and precision *>50%*), *precise* (recall *<=50%*, precision *>50%*), *complete* (recall *>50%*, precision *<=50%*) and *poor* (recall and precision *<=50%*). Only *5%* of all results lie in the excellent area, *10%* are precise, *15%* are poor and about *70%* are complete. That means, our system tends to optimize the recall.

Looking at the distribution of results reveals that the triangles form two clusters with *(recall, precision)* at *(80%,20%)* and *(10%,85%)*. That means, the low-level features produce extreme results with either high recall or high precision. The HWF results (circles) are about

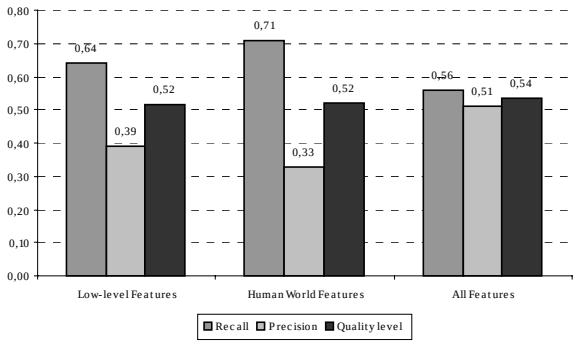

Figure 6. Quality comparison of evaluated methods. Using all features optimizes the quality level.

equally distributed in the poor and precise area. Most rhombs lie in the precise area with slightly better precision than the triangles. That means, using low-level features and HWF features together leads to more balanced results. Most results in the excellent area are rhombs.

Figure 6 summarizes the overall recall and precision (mean values over 100 tests each). The high-level features produce an excellent recall of *64%* with a poor precision of *39%*. The HWF features alone result in even more unbalanced results (*71%* and *34%*). Using all features reduces the recall but improves the precision.

The quality level in Figure 6 is the sum of recall and precision (for visualization it is divided by 2). It is a measure for the *maximum level* recall and precision can reach for a specific querying method and ground truth *independent from the query parameters*. The quality level for the method with all features is *54%*. This is a slight improvement of *2%* over the two basic methods. These results suggest that using HWF features refines the results of low-level features and balances the result set quality.

7 CURRENT AND FUTURE WORK

Next work on the HWF will include the development of more sophisticated versions of the descriptors and distance measures as well as additional tests on other image collections. In the future, we will try to base all HWF features on MPEG-7 image descriptors.

Additionally, we will define and investigate two further semantic feature layers: image creation artifacts (ICA) and chaotic image properties (CIP). ICA try to extract typical image errors that are originated in the photographing technology (digitized photos, video frames, etc.) or in the photographing task (shooting portrait photos, film scenes, etc.). For example, such a property could be color errors (derived from color histograms). These could be used to guess the age of an image. CIP extract chaotic elements of images (e.g. trees, flowers, etc.). They will be based on fractal theory and can be used to distinguish images of natural scenes.

8 CONCLUSION

In this paper we describe a novel approach to reduce the semantic gap problem of CBIR system. The basic idea is enhancing queries with high-level features that are based on low-level features. We have implemented a prototype for a feature class that describes human world properties. This feature class was tested in our test environment in 300 queries. The result was: using high-level features improves the quality of result sets by balancing recall and precision.

Our conclusion is that using semantic feature layers is reasonable when the used feature class suits the given querying problem (application domain). Otherwise it may even lead to a reduction of the querying performance. The semantic feature layer concept will be incorporated in the open VizIR project. Interested researchers are invited to join this project or use its results and deliveries for further CBIR research.

9 REFERENCES

[1] Breiteneder, C., and Eidenberger, H. A Retrieval System for Coats of Arms in Proceedings International Symposium on Multimedia Application and Distance Education (Baden-Baden Germany, 1999).

[2] Del Bimbo, A. Visual Information Retrieval. Morgan Kaufmann Publ., San Francisco CA, 1999.

[3] Eidenberger, H., and Breiteneder, C. A Framework for Visual Information Retrieval in Proceedings Visual Information Systems Conference (HSinChu Taiwan, March 2002), LNCS, Springer Verlag, 105-116.

[4] Furht, B., Smoliar, S.W., Zhang, H.: Video and Image Processing in Multimedia Systems. 2nd edn., Kluwer, Boston MA (1996).

[5] MPEG-7 standard documents Website. http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.htm

[6] Rational Unified Process Website. http://www.rational.com/products/rup/index.jsp

[7] Rui, Y., Huang, T., Ortega, M. and Mehrotra, S. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. IEEE Transactions on Circuits and Systems for Video Technology, 8/5 (1998), 644-655.

[8] Semantic Web Website. http://www.semanticweb.org

[9] Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., and Jain, R. Content-Based Image Retrieval at the End of the Early Years. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22/12 (December 2000), 1349-1380.

[10] Smith, J.R., and Chang, S.F. VisualSEEk: a fully automated content-based image query system in Proceedings ACM Multimedia (Boston MA, 1996), ACM Press, 87-98.

[11] Unified Modeling Language Website. http://www.uml.org

[12] Wood, M., Campbell, N. and Thomas, B. Iterative Refinement by Relevance Feedback in Content-Based Digital Image Retrieval in Proceedings ACM Multimedia (Bristol UK 1998), 13-20.