

An Approach to Visualize Image Retrieval Results

Anke Schneidewind Petra Neumann Ingo Schmitt

Faculty of Computer Science

Otto-von-Guericke-University, Magdeburg, Germany

E-mail: {schneidewind@iti|pneumann@|schmitt@iti}.cs.uni-magdeburg.de

Abstract

Many current content-based image retrieval systems suffer from poor result presentation. A sophisticated visualization can be used to identify differences between human perception of similarity and system-determined similarity. Analyzing such discrepancies is a prerequisite for a system trimmed towards the user's comprehension of the underlying retrieval process.

The aim of the visualization techniques presented in this paper is to provide a tool to analyze a mismatch between the user's perception and the system's calculation of similarity. We combine techniques of visual image retrieval and information visualization to acquire insight into the extracted feature data. In our project we implemented visualization techniques to present feature data on three different levels of abstraction. We discuss our experiences when working with a Data Table, a Parallel Coordinate Plot, and a Color Space Plot.

1. Introduction

The central problem of content-based image retrieval deals with the well-known semantic gap [18] between human perception of similarity and system-determined similarity computed from automatically extracted feature values. A human being operates on a high level of abstraction because s/he implicitly exploits knowledge about the image domain. In contrast, a computer system interprets images at a semantically low level. Given a query image, a retrieval system extracts feature values and applies some similarity or dissimilarity measures to find similar result images.

Many content-based image retrieval systems present their retrieval results as a simple list ranked by relevance to the query image. To alleviate the problem of the semantic gap the user can typically employ relevance feedback techniques to obtain a better result. Normally, the user hardly understands the system's way of 'perceiv-

ing' similarity and often s/he has to apply a trial-and-error method because s/he don't know how to adapt the query for a better result. Therefore, we need visualizations to assist the user in formulating queries.

In terms of making characteristics of the feature space more apparent some systems display result images in a two- or three-dimensional space [19, 15, 14]. One characteristic of these approaches is that they map the high-dimensional feature space down to a number of depictable dimensions.

This mapping is typically performed by feature clustering [7], multi-dimensional scaling [16], or by breaking the feature or distance values down to a few dimensions [15]. These techniques are also known from information visualization and text retrieval [22]. The information loss of these approaches, caused by dimension reduction, allows an abstract view of the global arrangement of result images.

We think that a specific insight in the system's way of computing similarity can assist the user in performing a more system-oriented feedback resulting in better retrieval quality. A step towards this aim is realized by blobworld [5]. The system shows the extracted average color of image regions. The system described in [3] that system highlights the regions in the result images which match the query. Both systems shows some details of computed (searched) feature in single result images.

Our idea is to intensify the use of information visualization techniques to improve result visualization. We propose to present both more abstract data using FastMap and more internal data showing detailed feature data. In this way the user gets an more extensive insight. This increases the possibility that the user finds the relevant features for a more exact query.

Our image visualization Tool (iVi-Tool) allows users to switch between views with differently abstract result visualizations: a simple Data Table, a Parallel Coordinate Plot [10], and a Color Space Plot. A higher view is always an abstraction of a lower view. In that way we are able to present more information than a simple result

list or the results of the above-mentioned approaches. To test our approach we create in a study queries which return undesirable results and tried to find the correct answer with the help of the visualizations¹.

Our results demonstrate that (1) our different kinds of visualizations help to discover information loss within the search process, (2) a human oriented visualization can be used to improve feature extraction, and (3) even an unsupervised user can gain important information detail for a next query.

In the following section we discuss techniques of information visualization and of content-based image retrieval. Section 3 introduces the implemented visualization tool with its particular components. In Section 4 we provide two scenarios demonstrating the advantages of our visual techniques in content-based image retrieval. Finally, in Section 5 we conclude by showing the significance of our work and give directions for further research.

2. Information Visualization combined with Content Based Image Retrieval

Information visualization techniques are about making data understandable and explorable so that the visualization helps users to gain knowledge about the data. It is the process of forming a mental model for the acquired data and so helping the viewer to understand underlying concepts, patterns, and connections within the data [22]. In this sense, information visualization is our method of choice in assisted content-based image retrieval.

A visualization is constrained by rules of effectiveness, expressiveness, and appropriateness [20]. These aims are sometimes contradictory. On the one hand, result visualization should be at an abstract level to prevent the user from bothering with too many system details and therefore, for an easy and fast interpretation. But on the other hand, visualization should reveal the system-determined notion of similarity. Further, we want to support experts and novices as well. Therefore, following the definition of the user-level pyramid for visualization [4] we created three views at different levels of abstraction for presenting retrieval results.

Interactivity is another major aspect in information visualization. We give the user the possibility to change the abstraction level by altering the amount of data displayed. Thus, the user is supported in gaining more insight into the underlying data and s/he will be assisted in developing a mental model of the data representing the query results.

¹The colored figures of the study results can also be found at <http://www.witi.cs.uni-magdeburg.de/~anschnei/forschung/>.

The visualization of large amounts of high-dimensional data has been an active research area in information visualization. Keim [11] gives a good overview and categorization of relevant visualization techniques. One of those is the well known *Parallel Coordinate Plot* [10]. The idea of parallel coordinates is to visualize high-dimensional points or vectors in a sequence of parallel axes. A vector v with values (v_1, v_2, \dots, v_m) is visualized as a polyline connecting the points v_1, v_2, \dots, v_w on m vertical axes. We use the Parallel Coordinate Plot to present high-dimensional feature vectors, see Figure 3(a).

Further, we use color icons [13] for our three-dimensional visualization in our *Color Space Plot*, see Figure 1.

A visualization cannot replace analytical calculations [20]. To allow calculations on the feature values we present a Data Table of feature data.

There are a number of CBIR systems which present more result information than just a list of images, of which some are already mentioned in Section 1. For instance Blobworld [5] shows average colors of image regions. 3D MARS [15] shows images as spheres to ease the search for clusters in a global database view. The image browser [16] uses multi-dimensional scaling to show different kinds of computable correlations between the feature vectors. A visualization of the retrieval process is realized by PicSOM [12]. It shows an underlying self-organizing map which is adapted by using relevance feedback mechanisms.

Clear connections between information visualization and content-based image retrieval are made by C. Chen. He uses a special technique [6] named *Pathfinder* to show connections between clusters of images in the database in a three-dimensional view. El Niño [19] uses distortion techniques like Fisheyes to browse large information spaces. In contrast to these approaches, we are not interested in browsing the database. Table 1 shows the differences between the visualizations of those approaches and of our approach.

So far, few approaches intend to assist the user in getting an understanding of the internal search process. The ones that do, support just one abstraction level. Furthermore, they often do not visualize single feature values. In addition, in some systems a user needs expert knowledge in order to understand the visualization (e. g., PicSOM).

3. The iVi-Tool

Our system is implemented on top of an Oracle database system and comprises an indexing and query component as well as a visualization tool. We operate on images and

visualization feature	iVi	Blobworld	3D Mars	image browser	pathfinder
no abstraction	+	-	-	-	-
low abstraction	+	+	+	-	-
high abstraction	+	-	-	+	+
global data view	-	-	+	+	+
result data view	+	+	-	-	-
view on single feature	+	+	-	-	-
view on feature space	+	-	+	+	-
view on cluster	-	-	+	-	+
image/cluster distances	+	-	+	-	-

Table 1. Comparison of different content-based image retrieval systems.

their respective feature values managed by the database system. The indexing and query component supports query formulation and fast algorithms for nearest neighbor search in high-dimensional space [1]. The visualization tool presents query results graphically and is implemented in Java employing the Java3D API.

After briefly introducing the index component we discuss our visualization tool along with its different kinds of data visualization.

3.1. Indexing and Query Component

The indexing component is responsible for extracting feature values from images stored in the database system. Taking the human perception into consideration we use the three-dimensional L^*a^*b color space [9, 21]. A further advantage of the L^*a^*b model is the separation of luminance from chrominance values. This allows, if necessary, to treat luminance differently from chrominance values. We partition the L^*a^*b color space into 1.500 subspaces. Each subspace is a bin of a color histogram. The allocation of the colors to the single axes L , a , and b is displayed at the bottom of Figure 3(a).

Let us assume that for every image with a resolution of $m \times n$ there is a function $i_L(x, y)$ which returns the luminance value, that is the matching interval number of the pixel at position (x, y) . Analogously, we assume the existence of the functions $i_a(x, y)$ and $i_b(x, y)$ returning the respective chrominance values.

The color histogram of an image is a function $h_{Lab}(L, a, b)$ which maps each subspace to the fraction of matched image pixels:

$$h_{Lab}(L, a, b) = \frac{1}{m * n} \sum_{x=1}^m \sum_{y=1}^n \begin{cases} 1 : & i_L(x, y) = L \wedge \\ & i_a(x, y) = a \wedge \\ & i_b(x, y) = b \\ 0 : & otherwise. \end{cases}$$

Due to the restrictions of presentable dimensions and appropriateness (see Section 2) we reduce the 1500 dimensions by constructing an L-histogram with 15 bins, and an a- and b-histogram each with 10 bins:

$$h_L(L) = \sum_{a=1}^{10} \sum_{b=1}^{10} h_{Lab}(L, a, b)$$

$$h_a(a) = \sum_{L=1}^{15} \sum_{b=1}^{10} h_{Lab}(L, a, b)$$

$$h_b(b) = \sum_{L=1}^{15} \sum_{a=1}^{10} h_{Lab}(L, a, b).$$

The 15 L-values in contrast to the respective 10 chrominance values shall allow a better search on gray-scale images. Concatenating these three histograms altogether provides a 35-dimensional vector for every image. These vectors are indexed using the AV-method, a high-dimensional index structure, proposed in [2].

As query type we restrict our considerations to a simple query by example. The user can upload a query image and specify the number of result images.

Our visualization, is conceptually independent from a specific, underlying distance function. For simplicity, we choose the euclidean distance.

3.2. Visualization Tool

Our visualization tool supports three different views on the result data. The views visualize the data on different levels of abstraction. The levels refer to the degree of graphical interpretation of the feature data. In an increasing order of abstraction we have:

1. *Data Table*: At the lowest level (because of no abstraction) we suggest a Data Table containing the L , a , and b feature values of the query result. Every row represents a feature vector. All rows are sorted by their similarity values. The Data Table is required, when a feature developer wants to obtain the feature data for further investigations.
2. *Parallel Coordinate Plot*: Each vertical line of the Parallel Coordinate Plot represents one dimension of the feature data and each polyline represents one feature vector. The plot is at a higher abstraction level than the Data Table level since the values are interpreted graphically, see Figure 3(a).

3. *Color Space Plot*: In this view at the highest level of abstraction the average color, the rank, as well as the dissimilarity value are graphically interpreted. The vectors are visualized as annotated spheres or thumbnails in a three-dimensional color space, see Figures 1, 2 and 3(b).

Each view contains a list of ranked result images. The user can interactively mark those images whose feature shall be visualized. Furthermore, the user can easily switch between views. To ease identification of images and their corresponding features the image borders and the visualized features are colored the same, see for example Figure 3(a).

The decision which visualization supports the best insight depends on the user's information need, his/her search goal, and background knowledge about the features. If the user wants to analyze the features in detail the choice will be the Parallel Coordinate Plot. For further calculations, feature data of the images can be taken from the Data Table. When the user needs a global understanding of the calculated similarity s/he probably chooses the three-dimensional Color Space Plot. All views provide an insight into the differences between human perception and the system's similarity from different perspectives.

Usually, the user starts at a high level of abstraction, for example with the Color Space Plot. From there, a switch to the Parallel Coordinate Plot visualizes the underlying feature data of selected vectors. The tangible feature data can be obtained from the Data Table. Thus, a user does not use one view exclusively but usually navigates through the views.

The interface of the Visualization Tool shows the returned images of the query in a linear order on the right hand side of the window. The images are equipped with thick border lines ranging from yellow to grey. Presumably most similar images according to the query image are marked by yellow colors and images which were returned as being most dissimilar are marked blue. This color range help the users to identify visualized objects in the Parallel Coordinate Plot.

In the following we will briefly introduce view 2 and 3.

Parallel Coordinate Plot

The Parallel Coordinate Plot visualizes the data from the Data Table in one view. The different feature histograms for luminance and chrominance are concatenated to one large diagram. The histograms of several images are visually superimposed, see Figure 3(a). Here the query image is a unicolored image.

The result images are related to their feature data by their color marks. The polylines are drawn in the same color that is used to draw the border around each image on the right side of the interface.

The Parallel Coordinate Plot allows a visual comparison of image data. It is useful to examine the influence of single features and feature values on the result distances. This helps users to perform feature based relevance feedback. Thus, the user can reweight the underlying distance function accordingly [17].

Color Space Plot

The Color Space Plot view presents feature data in a three-dimensional color space. Every image is mapped to a position within the color space. Depending on the underlying color space we distinguish the Lab3d view from the FastMap3d view.

Lab3d: As the name suggests we use the three dimensions of the L*a*b color space to present a three-dimensional result visualization. By computing the average color as a three-dimensional vector \bar{c}_{Lab} we assign to every image exactly one position in the color space:

$$\bar{c}_{Lab} = \begin{pmatrix} \bar{L} \\ \bar{a} \\ \bar{b} \end{pmatrix} = \sum_{L=1}^{20} \sum_{a=1}^{10} \sum_{b=1}^{10} \begin{pmatrix} L \\ a \\ b \end{pmatrix} * h_{Lab}(L, a, b).$$

It can be easily shown that the average color values can alternatively be derived from the individual L-, a-, b- histograms:

$$\bar{L} = \sum_{L=1}^{20} h_L(L) \quad \bar{a} = \sum_{a=1}^{10} h_a(a) \quad \bar{b} = \sum_{b=1}^{10} h_b(b).$$

In this color space we visualize every image of interest as well as the query image as a labeled sphere with the following parameters:

- *position*: The center of the sphere is determined by the average color.
- *color*: The color of the sphere equals the average color of the image.
- *label*: Every sphere is equipped with a label. The label presents the rank position and the ID of the corresponding image.
- *size*: The size of a sphere corresponds inversely to the distance to the query image. Similar images according to the query image are large spheres whereas dissimilar images are small spheres.

The sphere size s_i ranges from the first to the last result image between the predefined values $maxS$ and $minS$. The sphere size s_i for an image at rank position i is computed from its distance d_i :

$$s_i = maxS - \frac{(d_i - d_1) \cdot (maxS - minS)}{d_n - d_1}.$$

The value d_1 and d_n equals the nearest and the farthest neighbor distance, respectively. The variables $maxS$ and $minS$ are used to parameterize the size range.

Figure 1 presents a screenshot of the Lab3d view. On the right side the result images are listed in rank order. In order to identify or to compare single result images and the related spheres, the images and spheres can be visualized separately.

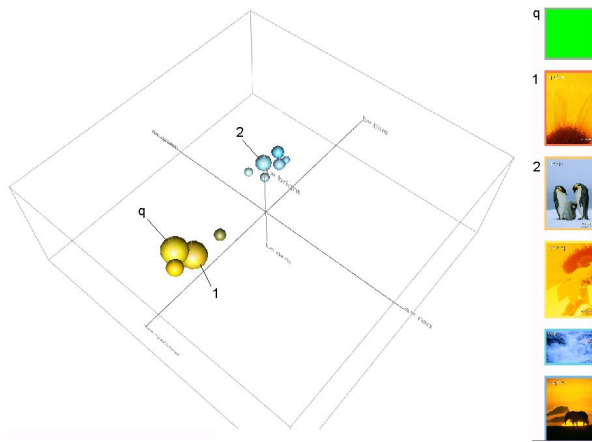


Figure 1. Screenshot from the visualization tool presenting the Lab3D view.

The novelty is, that the user gets three different kinds of internal informations at one sight without overloading the view:

- The user can see whether and how the calculated average color matches the user's perception of color.
- The user can see the relation between average color and calculated distance by examining size and color of similar images.
- The user see the matching of calculated distances and positions in the L^*a^*b space by position and size of the spheres.

Considering the relations between position and rank order the user can also make a statement on the extracted feature. As an example: if the average color of two images (spheres) are more similar than their rank order (positions), than there are some significant different single feature values.

The user can also switch from spheres to thumbnails of the corresponding images as demonstrated in Figure 3(b)a. With this view an unskilled user will get a more easy understanding of how the ranking is computed.

FastMap3D: FastMap, see [8], is an algorithm to map a metric to the euclidean space. After applying the map algorithm to a metric the resulting dimensions are descendingly sorted by their discriminating power to express the original distances. Therefore, if we choose to take the first three resulting FastMap dimensions to position a sphere in the Color Space Plot then these axes approximate the calculated dissimilarities relatively well.

By applying the FastMap algorithm we scale the 35-dimensional feature vectors down to three-dimensional vectors. Each of the three resulting axes can be seen as a linear combination of the histogram values. Therefore, instead of presenting the result images as spheres in the L^*a^*b space we position them within the FastMap space. Usually, the resulting FastMap axes do not correspond to a single color value. Similar to the Lab3d view the vectors can be presented as spheres or thumbnails, see Figure 2 of a later shown example.

Using this kind of visualization helps the user to identify latent dimensions which are mainly responsible for the computed dissimilarity values. The user gets an impression of how the images are arranged in space. Furthermore, s/he can compare the criteria of perceived dissimilarity with the axes of the FastMap space. A mismatch can help to find new feature values and other distance functions, which match the required perception of similarity better.

An interesting question is on which images the FastMap algorithm should be applied. So far, we assumed to take all result images. However, the user has alternatively the possibility to choose the images to be mapped. Thus, s/he can single out feature values with a good discriminatory power from feature values with a bad discriminatory power. For example, s/he can compare the FastMap dimensions from images which are acceptable with respect to the query with dimensions from unacceptable images.

A general advantage of the FastMap algorithm is the independence from original feature values and the distance function used. As a prerequisite we just need a metric. Therefore, this view is not restricted to original features representing visual properties.

This approach also has some drawbacks. It is sometimes hard to interpret the calculated FastMap axes and to relate them to definable or perceivable features. Furthermore, only the first three axes can be visually presented.

4. Example Scenarios

The next two scenarios will demonstrate how our proposed visualization tool can help to obtain better retrieval results.

In the first scenario we are simply interested to find images with dominating green color. Therefore, we take a plain light green query image. Unfortunately, the result images are mainly yellow. From the Lab3D view, we observe that the average color of the query image is more yellow than green (Figure 1). The reason is that the query image fell into the yellow interval although it is near to the green interval. Therefore, we get a better result by choosing a darker green for the query image (Figure 3(b)a). Nevertheless the new result shows red images on positions two and three. Examining the Lab3D view again we see that these images are positioned on nearly the same luminance value (vertical axis). Switching to the Parallel Coordinate Plot we can determine the significant values and out of these the relevant one, see Figure 3(a). Without relevance feedback, we can create a new query image taking the relevant luminance values out of the Data Table and combine them with the chrominance values of our query image to a new query. Another possibility is to take a result image that matches the combined query best. In our case we took the fourth image as new query because the Parallel Coordinate Plot of this image has no (single) high luminance values that can influence the result. The result of this query can be seen in Figure 3(b)b. It now suits our notion.

We obtain our right result by making our decision due to direct interaction with the data. Traditional relevance feedback systems will choose the significant features we can choose the relevant. Therefore we obtain the right result faster than those systems.

In future we plan to integrate relevance-feedback-mechanisms in our system. Thus, the last two manual steps will be omitted.

If a database developer examines Figure 1 he will see that the influence of the luminance seems to be generally higher than the human perception will rate it. Therefore, the right decision may be to reduce the number of luminance bins of the L^*a^*b -histogram, or use a general low weight for L. This will cause a higher influence of the human oriented chrominance values. Thus, the visualization can help in developing better feature vectors and distance functions.

In the second scenario we started with the query image seen in the right upper corner of Figure 2. We switch to the Lab3D view. The high similarity between the average color of the spheres causes them to stick together. However, the FastMap3D view automatically finds the right axes to discriminate good from bad result images.

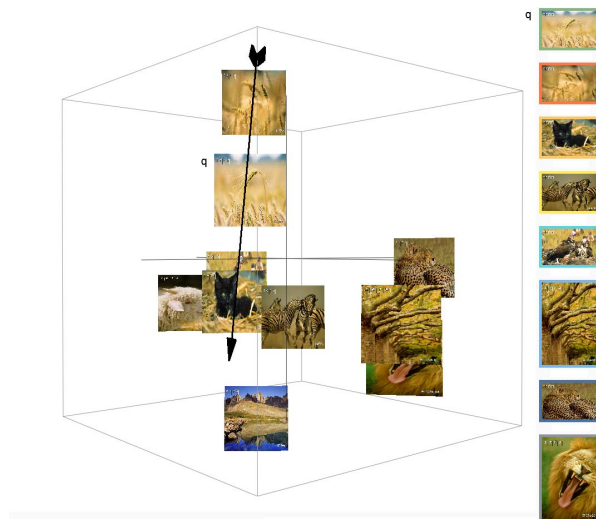


Figure 2. Color Space Plot FastMap3D.

We can determine an artificial axis which meets our notion of similarity best, shown as an arrow in Figure 2. Employing the information of the artificial axis the system can determine which combination of original feature values is mapped to that axis. With this information we are able to extract new features in order to retrieve better retrieval results.

5. Conclusion and Outlook

We presented three different visualization on different levels of abstraction and discussed their contribution for understanding the retrieval result. Furthermore, we demonstrated their potential to unveil discrepancies between the user's and the system's notion of similarity.

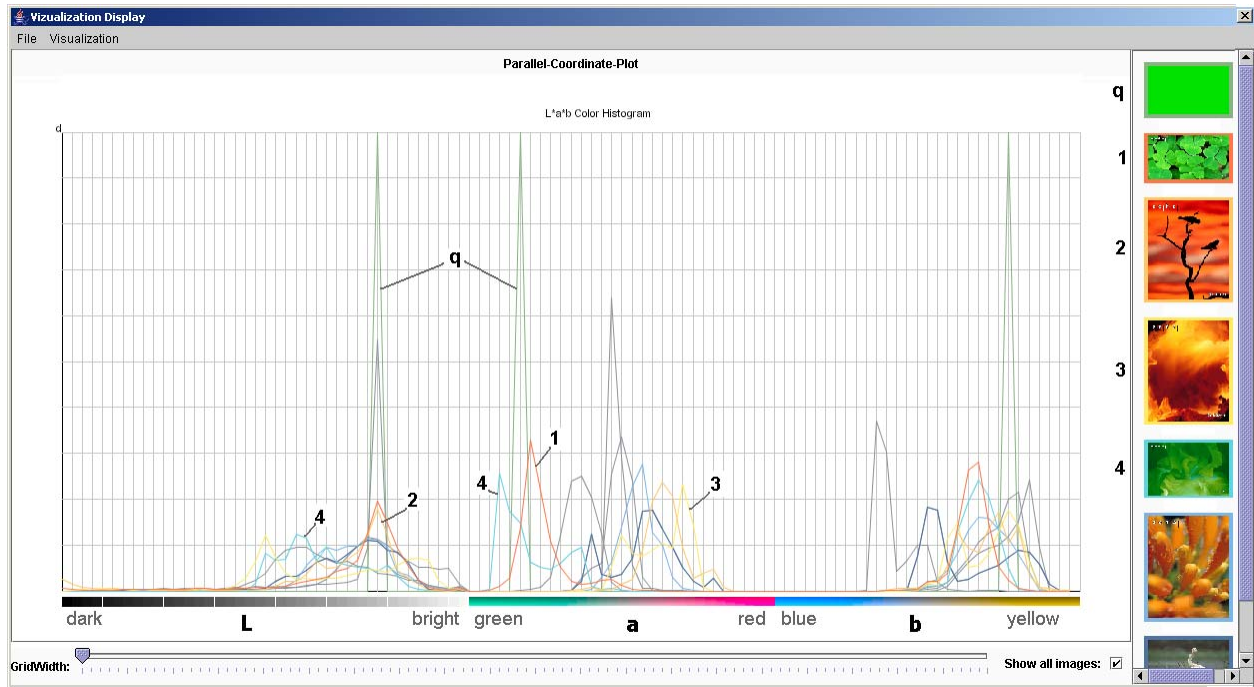
The novelty can be found in the combination of the approaches of image retrieval and information visualization. We use abstract visualizations to give the user an overview of the relations among the result images and detailed views to reveal significant differences among single feature values. The possibility to switch between the views also shows the relations between the abstract data and the single values. The visualizations are created in such a way that the user is led from the view of high abstraction to the details of the views with a lower abstraction. In that way we make it easy to interpret the Parallel Coordinate Plot because the user knows where he have to search in the plot. Contrary to other systems we give the user tools to interact and examine the pure feature data. Therefore the user can estimate the relevance of a single feature value on its own. To make our work complete we will conduct a user study. The next step is to use visualization results to trim automati-

cally the retrieval system to retrieve better result images. Therefore, we will investigate the impact of our results to a feature-oriented relevance feedback algorithm. A next aim is to test other features than color histograms, for example texture features. In those cases the Lab3d view is of no use and we have to create special three-dimensional visualizations for each feature or we confine the examination on the FastMap view and the Parallel Coordinate Plot.

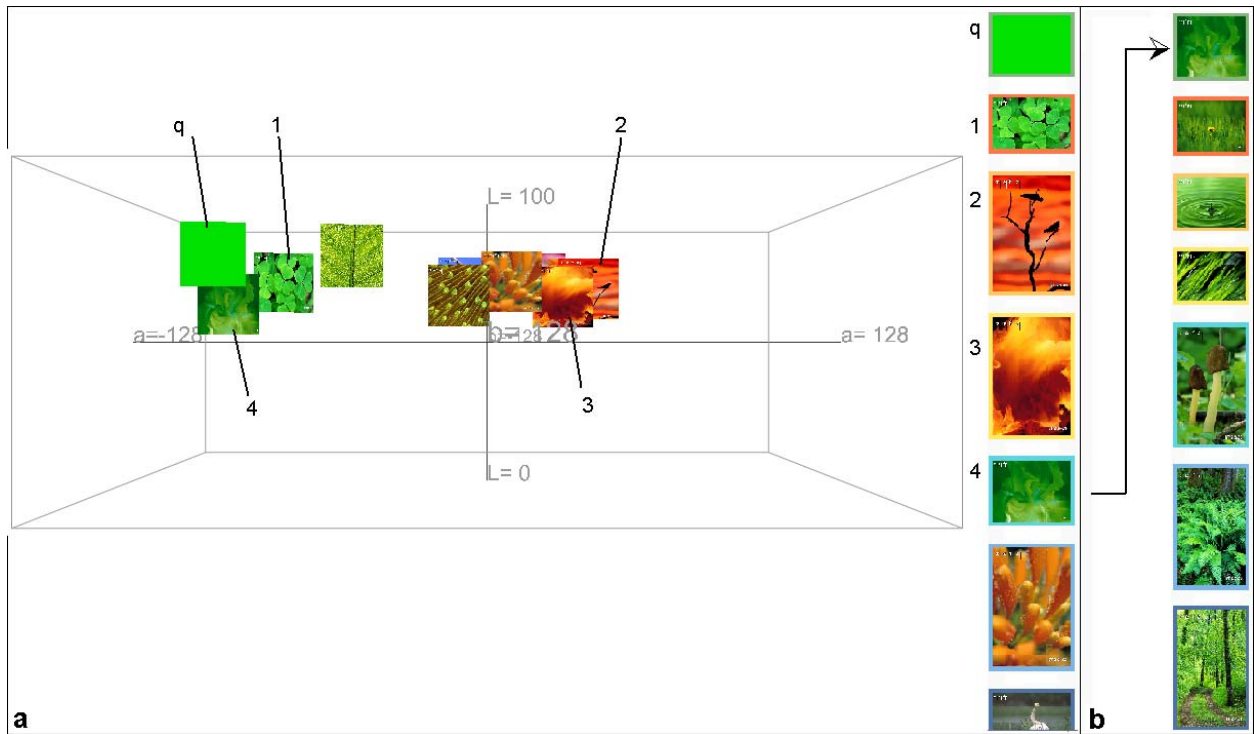
Furthermore, we will concentrate on developing helpful visualization techniques for complex queries.

References

- [1] S. Balko. *Grundlagen, Entwicklung und Evaluierung einer effizienten Approximationstechnik für Nearest-Neighbor-Anfragen im hochdimensionalen Vektorraum*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, 2004. in German.
- [2] S. Balko, I. Schmitt, and G. Saake. The Active Ver-tice Method: A Performant Filtering Approach to High-Dimensional Indexing. *Elsevier Data and Knowledge Engineering*, 2004. to appear.
- [3] H. G. P. Bosch, A. van Ballegooij, A. P. de Vries, and M. L. Kersten. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 513–516, Tokyo, Japan, August.
- [4] J. R. Brown, R. Earnshaw, M. Jern, and J. Vince. *Visualization: using computer graphics to explore data and present information*. John Wiley & Sons, Inc., 1995.
- [5] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *Third International Conference on Visual Information Systems*, pages 509–516. Springer, 1999.
- [6] C. Chen. *Mapping scientific frontiers: the quest for knowledge visualization*. Springer, 2003.
- [7] H. Chen, G. Gagaudakis, and P. L. Rosin. Content-Based Image Visualization. In *The Proceedings of the IEEE International Conference on Information Visualization (IV'00)*, pages 13–18, 2000.
- [8] C. Faloutsos and K.-I. Lin. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In M. J. Carey and D. A. Schneider, editors, *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 163–174, San Jose, California, 22–25 1995.
- [9] D. D. Hoffman. *Visual intelligence: how we create what we see*. Norton & Company, Inc., New York, 1998.
- [10] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st conference on Visualization '90*, pages 361–378. IEEE Computer Society Press, 1990.
- [11] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78, 2000.
- [12] J. Laaksonen, E. Oja, M. Koskela, and S. Brandt. Analyzing low-level visual features using content-based image retrieval. In *In Proceedings of 7th International Conference on Neural Information Processing (ICONIP 2000)*, pages 1333–1338, Taejon, Korea, November 2000.
- [13] H. Levkowitz. Color icons: merging color and texture perception for integrated visualization of multiple parameters. In *Proceedings of the 2nd conference on Visualization '91*, pages 164–170. IEEE Computer Society Press, 1991.
- [14] B. Moghaddam, Q. Tian, and T. S. Huang. Spatial visualization for content-based image retrieval. In *International Conference on Multimedia and Expo (ICME'01)*, Tokyo, Japan, Sept. 2001.
- [15] M. Nakazato and T. S. Huang. 3D MARS Immersive Virtual Reality for Content-Based Image Retrieval. *Proceedings of 2001 IEEE International Conference on Multimedia and Expo (ICME2001)*, pages 22–25, August 2001.
- [16] Y. Rubner and C. Tomasi. *Perceptual Metrics for Image Database Navigation*, chapter "Feature Similarity". Kluwer Academic Publishers, Boston, December 2000.
- [17] Y. Rui, T. S. Huang, and S. Mehrotra. Relevance feedback techniques in interactive content-based image retrieval. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 25–36, 1998.
- [18] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41:288–297, 1990.
- [19] S. Santini, A. Gupta, and R. Jain. Emergent semantics through interaction in image databases. *Knowledge and Data Engineering*, 2001.
- [20] H. Schumann and W. Müller. *Visualisierung: Grundlagen und allgemeine Methoden*. Springer, Berlin, 2000.
- [21] M. W. Schwarz, W. B. Cowan, and J. C. Beatty. An experimental comparison of rgb, yiq, lab, hsv, and opponent color models. *ACM Trans. Graph.*, 6(2):123–158, 1987.
- [22] R. Spence. *Information visualization*. Addison-Wesley, ACM press books, Harlow, 2001.



(a) Parallel Coordinate Plot.



(b) Searching for green images in the Lab3d view.

Figure 3. Visualizations in the iVi tool.