Proc. of 4th Int. Conf. on Visual Info. Systems, Lyon, France, 2000

Integrated Browsing and Searching of Large Image Collections *

Zoran Pečenović[1,2], Minh N. Do[1], Martin Vetterli[1], and Pearl Pu[2]

[1]Laboratory for Audio-Visual Communications
[2]Database/Human Computer Interaction Laboratory
Swiss Federal Institute of Technology Lausanne
CH-1015 Lausanne, Switzerland
{Zoran.Pecenovic, Minh.Do, Martin.Vetterli, Pearl.Pu}@epfl.ch

Abstract. Current image retrieval systems offer either an exploratory search method through browsing and navigation or a direct search method based on specific queries. Combining both of these methods in a uniform framework allows users to formulate queries more naturally, since they are already acquainted with the contents of the database and with the notion of matching the machine would use to return results. We propose a multi-modes and integrated image retrieval system that offers the user quick and effective previewing of the collection, intuitive and natural navigating to any parts of it, and query by example or composition for more specific and clearer retrieval goals.

1 Introduction

1.1 Searching and Browsing

The online availability of huge collections of images in digital form requires effective and efficient tools that allow users to search, browse and interact with such databases. There are two principal paradigms for retrieval from large image databases: direct search and browsing (or serendipitous discovery).

In the direct search environment, users present the system with a query (which can be an image, selected or constructed by painting and composition, or a keyword, or color characteristics of an image) and ask for the system to look for "more similar images" that match the query. One of the problems with direct search is that even with recursive query refinements and relevance feedback, users can be trapped in a small group of undesirable images. This fact arises from the typical large size of the search space and the local nature of any similarity measurements. Naturally, in this scheme, systems don't provide alternatives or images apparently not matching the user submitted query, thus forcing the user to slowly explore a small vicinity of the initial query. An additional problem is the construction of queries themselves as it can prove to be non-trivial for many users.

* This work was supported by the Swiss National Fund for Scientific Research, Grant no. 21-52439.97.

At the beginning of his interaction with an Image Retrieval System (IRS), the user is unable to know all of the data contained in the database, nor its structuring. Furthermore s/he can not know whether the desired type of data is present in the collection at all. A similar scenario occurs when a novice user starts using the system and feels insecure about the system's ability to evaluate similarity, or when the desired information is not clearly defined in the user's mind. In these cases, the browsing environment allows users to start from an overview of the collection and iteratively "zoom in" on the interesting parts until they can locate the desired image. Furthermore, the gathered images during the browsing process can be used as initial seed images to be modified into queries for finer retrieval using the search tools.

While the search environment is more popular and is extensively used in conventional retrieval systems, notably for *text retrieval*, the situation could be different for *image retrieval*. The reason lies in the fact that unlike in the text case where document relevance is difficult for users to grasp quickly, the relevance of an image to a specific query can be judged almost at a glance. Hence browsing and searching are complementary and necessary tools for effective image retrieval systems. In addition they share and rely on the major components of image retrieval systems: "good" extracted features from images and "good" similarity measurement between two images. While these two tasks are still open problems for study, we focus here on how to enhance the usability of underlying features and measurements and how to add an extra layer of interaction with image databases integrating both search and browse paradigms.

The main problem with image browsing is how to produce a visualization of the whole collection (or part of it) and how to provide an effective mechanism to navigate through that image database. One way of constructing the visualizing environments is to project images into a two or three-dimensional space where similar images are close to each other [7]. Experiments show that users preferred to view retrieved results in 2-D maps rather than in ranked lists by similarity in a "reading-order" where adjacency among retrieved images has little meaning. The operations on this projection map via continuously zooming and panning provide users with maximum freedom in navigating through the collection. However, without structure this becomes a burden to view and to display. Some systems proposed to organize images into hierarchical structures like self-organizing maps [5] or pyramids [1]. However in those systems, navigation is restricted to only "discrete" steps since images are positioned in tree-structured grids.

1.2 System Overview and Paper Organization

In this paper we will present our Content-based Image Retrieval and Consultation User System (CIRCUS), which offers the best from both worlds with seamless switching between modes. The main idea behind our approach is that dynamic and interactive visualizations of the data, *combined* with direct searching by queries, will help the user retrieve the desired information.

The typical scenario for a user of CIRCUS is the following: first the user would consult the tree structure for the overview of the image collection. For any particular interesting region they can use the browsing facility to pan or zoom in for more detail and see more images around that region. Finally, those images can be used as examples or can be modified to present as a new query in the search mode. In addition our system allows the incorporation of user's relevance feedback to adjust the similarity effectiveness.

In the next section, we describe the underlying building blocks of our system. Section 3 describes the visual interface taking advantage of those structures together with some examples. Section 4 provides some evaluation results together with discussion and outlook.

2 Underlying Building Blocks

The following computational tasks are performed *off-line* on image databases:

1. Extraction of d features from each of the n images. Those features allow similarity measurements between images to be computed.
2. Construction of a visual map of the image collection by projecting n feature vectors from $\mathbb{R}^d$ into 2-D spaces.
3. Hierarchical clustering of images and assigning representative images to result clusters.

For the completeness, we now briefly describe each of those building blocks.

2.1 Feature Extraction and Similarity Measurements

Our system is based on color features via color histograms [10] and moments [9], texture features via energies of the wavelet decomposition [8], and shape features via wavelet maxima moments [2]. Those features are normalized into a common range using deviations from medians of each component. If we denote the j-th feature from the i-th image by $\tilde{x}_i^j$ and n is the number of the images in the database then the normalization is performed by computing

$$x_i^j = \frac{\tilde{x}_i^j - m^j}{s^j}, \tag{1}$$

where m^j is the median of $\{\tilde{x}_1^j, \tilde{x}_2^j, \ldots, \tilde{x}_n^j\}$ and $s^j = \frac{1}{n}\sum_{i=1}^{n} |\tilde{x}_i^j - m^j|$. Other popular normalizations replace m^j and s^j by the mean and standard deviation of the feature j, respectively. However those are known to be more affected by *outliers* — the features which due to uncontrollable causes deviate from the normal range.

Once the features are normalized, Euclidean distances on $\mathbf{x}_i$; $i = 1, 2, \ldots, n$ are used to estimate the dissimilarity between images.

2.2 Multivariate Data Projection

The purpose of this step is to construct a visualizable map of images so that adjacent images are similar. Given images can be indexed by feature vectors in a high dimensional space, this task amounts to a multivariate data projection. There are many methods for accomplishing this including: Principal Component Analysis (PCA), Sammon's projection (SP), and multidimensional scaling [4]. The SP is an non-linear projection method and is shown to be more adaptive to complex data sets so it is chosen in our system.

Given a set of n vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ in $\mathbb{R}^d$, Sammon's method attempts to find a set $Y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$ in a lower dimensional projected space $\mathbb{R}^p$ (typically $p = 2$) such that the distance between pairs of vectors in X are preserved in their images in Y. Let us denote d^*_{ij} the Euclidean distances between $\mathbf{x}_i$, $\mathbf{x}_j$ in the original space and d_{ij} the distance between their images $\mathbf{y}_i$, $\mathbf{y}_j$ in the projected space. Sammon's algorithm tries to minimize the following error term:

$$E(Y) = \frac{1}{\sum_{i<j} d^*_{ij}} \sum_{i<j} \frac{(d^*_{ij} - d_{ij})^2}{d^*_{ij}} \tag{2}$$

Minimization of $E(Y)$ is an unconstrained optimization problem of $n \cdot p$ variables y_i^j $(i = 1, \ldots, n;\ j = 1, 2, \ldots, p)$. Sammon's algorithm uses a gradient descent method to reconfigure Y so as to minimize $E(Y)$ in an iterative fashion.

One problem with the original SP is that it does not offer the generalization ability. When new images are inserted into the database, determining their positions in the existing map would require a re-run of the SP for the whole collection! For this, [6] proposed an interesting solution by modeling Sammon's projection using neural networks. In their model, the gradient descent optimization is converted into a back-propagation training.

Another problem with Sammon's projection is the large number of variables to optimize so that a good initial configuration is required to speed up convergence and to avoid local minima. To facilitate this, we first use a PCA projection map (which is linear transformation and can be computed relatively fast) as an initial configuration and then apply Sammon's iterations to refine the map.

2.3 Image Clustering

With the result of projection of feature sets into a 2-D space, images can be positioned in a visualization map. Our interface allows the user to continuously pan across this plane and zoom in on any particular regions (see Sec. 3.2). However with large collections, actual images only become visible when the user zooms in on a very small part of the map. Therefore we create an extra facility to organize images into hierarchical tree-structures.

To accomplish this we recursively cluster images in the higher dimensional *feature space* into regions that contain similar images. For each found region, we pick a representative image that is closest to the centroid of the cluster. This tree structure helps users efficiently navigate through the image collection.

At each level, the clustering step is done via a *K-means* algorithm with successive splitting of the centroids as in the LBG vector quantization [3]. Furthermore we enforce the constructed tree to be balanced by adding a heuristic constraint on the size of the clusters [1].

3 Interactive Visualization for Searching & Browsing

As hinted in the introduction, and relying upon the available retrieval and data structuring methods described in the previous section, we now present an integrated browsing & searching user interface.

3.1 The Searching Mode

The basic user interface of CIRCUS provides several query construction mechanisms along with several session management and data marking utilities. The primary query paradigm in most IRS's is the query by example — more complex scenarios require more complex queries. Our current experimental system can cope with the query types detailed below.

Query by Example (QbE) The user selects several example images (positive and negative for accommodating relevance feedback) and asks the system to retrieve similar ones (see Fig. 3 in Sec. 3.3).

Query by Text (QbT) The user can also specify keyword annotation either required to be present in the image or simply as additional desired property of the result. This type of query can be executed at any time by using the adequate field in the main tool-bar.

Query by Color (QbC) As a third possibility s/he can also construct a query specifying the color properties of the sought images, like proportions and general hues (see Fig. 1 below).

Query by Painting (QbP) Using the final and most flexible query tool, s/he can modify existing images or sketch new ones using elementary drawing, coloring and montage tools for copying & pasting. This new image can then be used as the example for a QbE (see Fig. 1 below).

Combined Query (CQ) Finally a query can be constructed using any combination of the above tools through operators familiar to most users like "OR", "AND" & "AND NOT". These queries are processed if possible jointly, if not then separately, and the results combined by the IRS into a single set.

In the searching mode, the user can operate with a set of tools to express the desired properties of the result. It is primarily adapted to users with a good knowledge of both the system and the underlying collection. Furthermore users must have a clear idea of what is being searched for, so as to express their needs in a precise way. The direct search mode is not adequate for explaining to the user the notion of similarity and match implemented in the IRS. Starting from these limitations we introduced the browsing mode[1] which is detailed below.

[1] A simple browsing of the collections by random samples is also available as part of the direct-search mode.

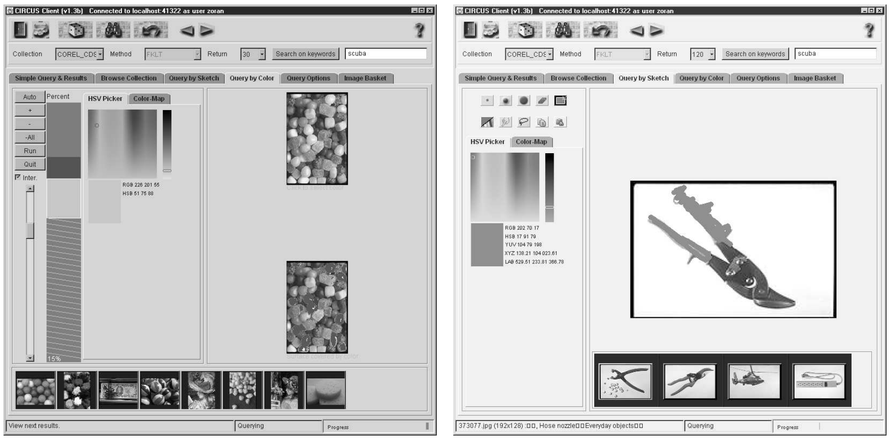


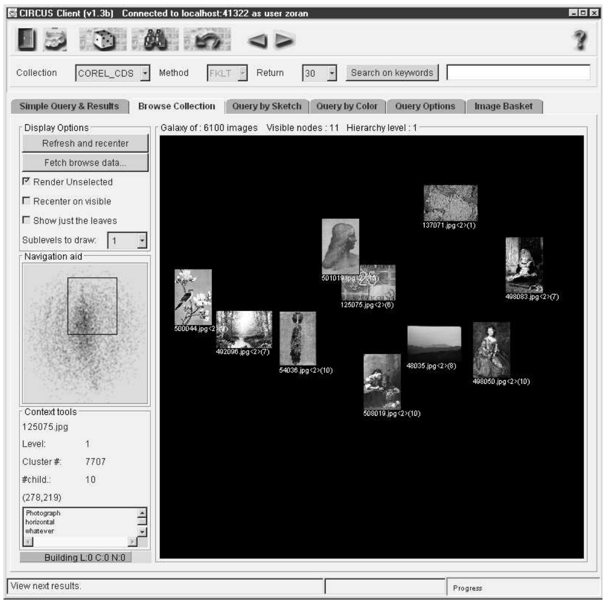
Fig. 1. The elementary query construction tools and some results. Left: (QbC). The proportion of several colors (picked from a palette or from existing images) was specified by dragging the colors in the left vertical area, at the bottom of the screen the user sees dynamically several results. Right: (QbP). The drawing area shows a simple sketch, with the user painting the pliers handles in a different color. Again a couple of best matches are shown below the drawing area, as the user draws.

3.2 The Browsing Mode

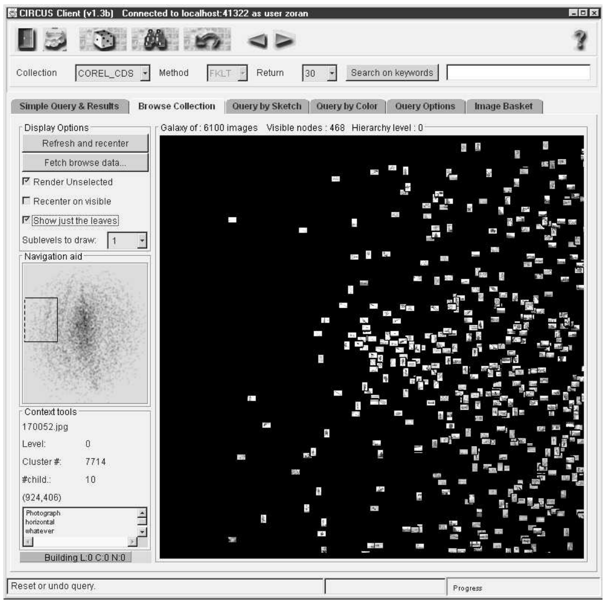
As a solution to the above expressed concerns, we argue that a fully interactive *real-time* display of a hierarchically clustered collection, projected into a two dimensional space can bridge the gap between the user and the system. By doing this we enhance the user's confidence, query specification ability and degree of acquaintance to the system. The human perceptual system offers a very high bandwidth channel for communication, and today the computing power on everyone's desktop allows for un-precedented possibilities for interaction. The demonstrated system runs as a JAVA application/applet on a Pentium II, 128Mb RAM system.

Based on the pre-computed results of the feature projection and hierarchical clustering steps detailed in Sec. 2, we display the 2-D browsable space in which images are presented using a semantic multi-resolution approach (Fig. 2). When examining the hierarchy at any level, the users see the representative images of each cluster displayed at the projection of the cluster centroid. Optionally several sub-levels can be displayed along with currently examined level; these sub-clusters are then rendered in a smaller size than their parents. The users can then zoom in, or pan around the display to bringing more images into focus or enlarging the images.

The display is linked with a progressive multi-resolution coding of the images, allowing the display of detail only when it becomes really necessary. It is also coupled with a semantic zoom both for single images and for clusters, i.e. when the zoom level becomes large that the images' symbolic annotation and meta-



The user is descending the hierarchy tree and has reached an interesting cluster. By panning around s/he can view the cluster's neighborhood. By clicking on an image s/he can descend to the next level of hierarchy, or by clicking outside of an image climb to the parent level. All transitions from one view to an other are animated, by automatic panning and zooming.



The user is examining the entire collection at a given level of the hierarchy (leaf level in this case). S/he can navigate through a galaxy, and — as far as the system resources permit it — images are rendered as thumbnails, if their number increases too much the system shows them as colored dots.

Fig. 2. Browsing the collection. In each example, the navigation aid — the overview map on the left with a small box indicating the current viewed region — helps the user know in which part of the space s/he is.

data becomes visible, and as the user moves in closer to a cluster, its sub-levels are automatically displayed. By clicking on an image, the display centers around it and if the image represents a cluster, it's sub-clusters are displayed. This multi-resolution scheme ensures for fast updating of the display, without ruling out the users ability to access any of the data available. At all ties, while interacting with the main display canvas, a smaller version of the entire space is represented in a side view with only the currently visible area highlighted. Consulting this view, the user can always know what part of the search space s/he is examining.

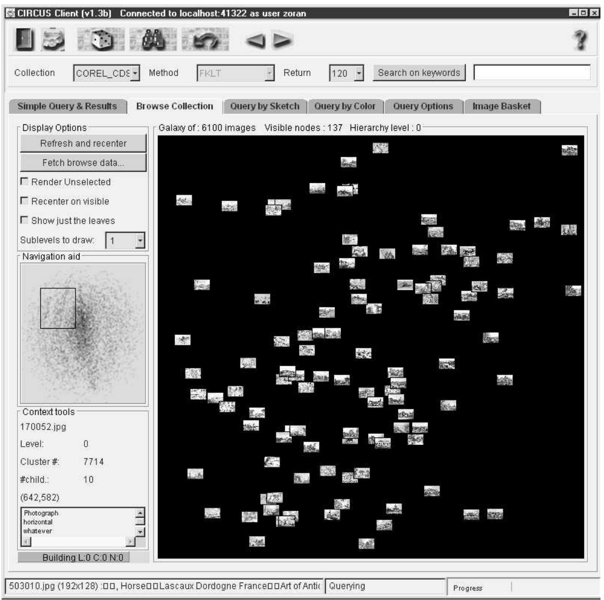
3.3 Seamless Integration

The integration of the searching mode with the browsing mode is achieved mainly through two mechanisms: first by a tight coupling of the browsing display with the query results, and second by direct manipulation and query specification in browsing mode. These two mechanisms allow users to browse and to perform different direct searches without ever leaving the browsing mode screen.

The first mechanism actually allows users to filter out the information displayed in the galaxy display using the results of the previous query or sequence of previous queries. For example if the user has issued a specific query on images containing the keyword "horse" and has given one example image from a previous search, the galaxy will show only the clusters containing images that satisfy those queries. The other images will be hidden (or grayed out). Thus iteratively, after specifying several queries, the size of the result set will be reduced and the display successively pruned of the undesired images and narrowed down to the desired ones. Figure 3 illustrates this principle by showing the restricted set of images returned by the described query as well as the QbE results in the linear-list "reading order" interface correspondingly.

The second mechanism allows users to specify new query constraints and new queries, directly from the galaxy display. For instance if the user right-clicks on an image in the display, a context-dependent menu allows him to launch a QbE on that image. Similarly, if the image is displayed with enough detail, a set of drawing tools appears in the context tools panel, allowing the user to paint directly in the galaxy display. This is depicted in Fig. 4. At any moment, the user can enter additional textual information for retrieving other images or restricting the already returned results. We believe that this *immersive* approach, allowing the user to navigate a space where the images "live" grouped by similarity, interact with them to specify new queries and immediately perceive the implications of his actions on the results is essential in an effective interactive IRS. Further allowing the user to change his mind, experiment "what-if" scenarios, and examine the results in a uniformly employed image space, helps her/im construct more intelligent future queries.

Viewing the results of a query by example in a ranked list displayed in "reading-order".



The same results displayed in the browsing view.

The actual query could have been initiated and can be refined from any of the two displays.

Fig. 3. Tight coupling of the browsing and searching facilities.

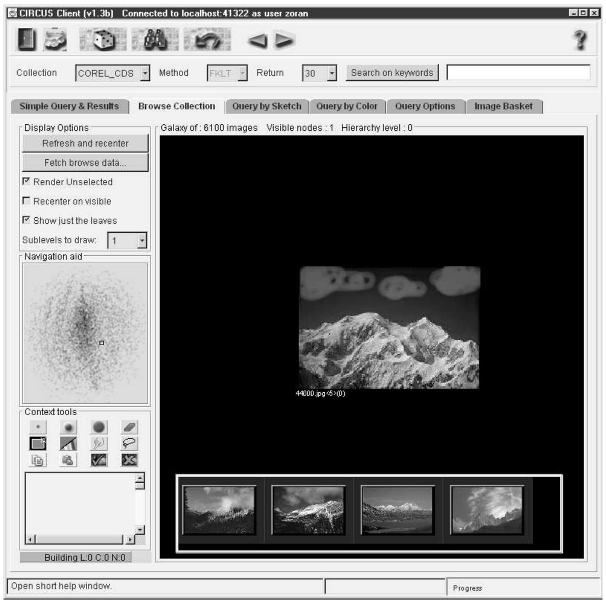


Fig. 4. Second integration mechanism. The user has zoomed in far enough to see only one image on the screen, the context tools then allow him to paint some "clouds" on the top of the original image, and the system responds with some candidate results.

4 Evaluation, Discussion & Outlook

Although we could not perform full scale usability testing, we did perform some experiments with several novice (4) and occasional users (2). We outline below some of the major conclusions.

In all of our experiments, the users had received a description of the task to perform and were observed during execution. Their comments and reactions were recorded. The major difficulty that was revealed was the discovery of the functionality (navigation, coupling) without prior explanation. Once acquainted with the controls, most users found the navigation natural and all of them found it beneficial to their search task. The more experienced users found the benefit to be lesser; justifying it by claiming that they already were familiar with the system's reactions and the database. This is by no means a surprise; it confirms our assumptions in Sec. 1.1. The novice users, on the other hand, grasped the idea of similarity, judging it somehow artificial (as one would expect). They greatly appreciated the synthetic overview the system was giving of the database comparing it to category-hierarchy search engines in text retrieval. Due to the lack of available testers and clear tasks associated with ground truth results, we could not perform comparative tests to measure the user performance increase when using the integrated browsing and searching mode. The information need clarification aspect we claim could thus not be investigated.

In brief, we have shown a working integrated system that enhances the user's ability to interact with an image retrieval engine. The two principal retrieval modes: searching and browsing have been merged into a seamless interaction model. The user thus gains insight in the organization of a large collection and can clarify his information needs and formulate more precise queries. The semantic navigation through a hierarchical representation of the data and the multi-resolution display (both for the image "signals" and their meta-data) enhance simultaneously the effectiveness and the speed of our system. The bottom line is that user satisfaction and ability to communicate with the machine are drastically enhanced.

We plan to solve some issues still pending, namely: (a) the enhancement of the map generation algorithms so as to accommodate projections and assignments of new images not yet in the collection; (b) increase the coupling between the search and browse modes, especially by integrating textual data into the browsing process; (c) enhance the multi-resolution rendering of the system, in the framework of slow communication channels between the IRS and the client applet.

References

[1] J. Chen, C.A. Bouman, and J.C. Dalton. Similarity pyramids for browsing and organization of large image databases. In *Proc. of SPIE/IS&T Conf. on Human Vision and Electronic Imaging III*, volume 3299, pages 563–575, 1998.

[2] M. Do, S. Ayer, and M. Vetterli. Invariant image retrieval using wavelet maxima moment. In *Proc. of 3rd Int. Conf. in Visual Information and Information Systems*, pages 451–458, 1999.

[3] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Boston, MA, 1992.

[4] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

[5] J. Laaksonen, M. Koskela, and E. Oja. Content-based image retrieval using self-organizing maps. In *Proc. of 3rd Int. Conf. in Visual Information and Information Systems*, pages 541–548, 1999.

[6] J. Mao and A. K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Trans. on Neural Networks*, 6:296–317, March 1995.

[7] Y. Musha, Y. Mori, A. Hiroike, and A. Sugimoto. An interface for visualizing feature space in image retrieval. In *Machine Vision and Applications*, pages 447–450, 1998.

[8] J. R. Smith and S.-F. Chang. Transform features for texture classification and discrimination in large image databases. In *Proc. of IEEE Int. Conf. on Image Processing*, 1994.

[9] M. Stricker and M. Orengo. Similarity of color images. In *Storage and Retrieval for Image and Video Databases III*, volume 2420 of *SPIE*, pages 381–392, 1995.

[10] M. Swain and D. Ballard. Color indexing. *Int. Journal of Computer Vision*, vol. 7(1), 1991.