Technologie-Zentrum Informatik

# Technical Report 29

## Similarity Search by *PictureFinder*

Thorsten Hermes, Andrea Miene & Otthein Herzog

TZI-Bericht Nr. 29
2003

Universität Bremen

**TZI-Berichte**

Herausgeber:
Technologie-Zentrum Informatik
Universität Bremen
Universitätsallee 21-23
28359 Bremen
Telefon: +49-421-218-7272
Fax: +49-421-218-7820
Email: info@tzi.de
http://www.tzi.de

Technical report

# Similarity Search by *PictureFinder*

Thorsten Hermes, Andrea Miene & Otthein Herzog

TZI-Bericht Nr. 29, 2003

**Organisation:**
Thorsten Hermes, TZI - Center for Computing Technologies, University of Bremen

# Similarity Search by *PictureFinder*

Thorsten Hermes, Andrea Miene and Otthein Herzog
Image Processing Department
Center for Computing Technologies - University of Bremen
Universitaetsallee 21-23
28359 Bremen, Germany
email: $\{hermes|andrea|herzog\}$@tzi.de

December 2003

**Abstract**

The text searching paradigm prevails over the internet even when users are looking for image data. The current state of the art is searching for annotations that have been made manually. When annotations are left empty, which is usually the case, searches on the image file names are performed. The graphical search paradigm, searching image data by querying graphically, either with an image or with a sketch, currently seems not to be the preferred method partly because of the complexity in designing the query. In this paper, we present our approach to and our first experimental results on graphical queries which were performed with PictureFinder. Furthermore, we describe the procedure of indexing and the methods used. Furthermore, we are bridging to region-based image retrieval (RBIR). PictureFinder is a system which currently supports "full image retrieval" in analogy to full text retrieval.

**Keywords**: Image Processing, Image Retrieval, Similarity Search, Query by Example

## 1 Introduction

Imagine a film director directing a shot only by a verbal description without seeing the surrounding context. Or consider for example a painting of Rene Magritte, *The Domain of Arnheim*: the painting communicates much more than words can say. Images should be seen in combination with their visual characteristics and therefore textual queries should be combined with graphical ones. Thus, one can search for images by their style or the appearance of certain objects [SWS+00] or by their visual characteristics like certain color, texture, and contour.

This simple few notions lead to the three following aspects, which are first more concrete and second have mutual relations to our approach and vice versa.

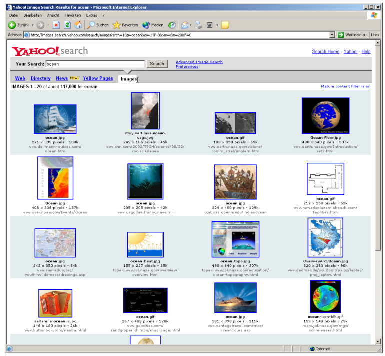

Figure 1: "Ocean" textual query and corresponding results of Yahoo

First, the World Wide Web as a possible concrete application domain is considered. Second, the influence of the recent research branch region-based image retrieval (RBIR) is addressed, and a last but not least third the relations to the very interesting research field in the context of content-based image retrieval (CBIR) the similarity search is contemplated.

## 1.1 Web

Consider images on the World Wide Web. Querying certain images by a verbal description (annotation) can lead to surprising results because some annotations consist only of the image file names. Try to search the term "ocean pictures" with e.g. Yahoo[1]. Some example results are displayed in figure 1. The images seem to be the result of a verbal query based only on the image file names.

This is why a graphical query of visual image characteristics like a certain color, texture, or contour in combination with a textual/verbal annotation is necessary. It is important for a promising site in the Web that the search engine provides the results expected by the user. Therefore it is essential that one can search for the certain picture one has in mind.

PictureFinder is a system for the automatic annotation of large image databases concerning the visual characteristics and it offers a successful image search.

## 1.2 Region-based Image Retrieval (RBIR)

RBIR was recently proposed as an extension or a branch of CBIR. An RBIR system automatically segments images into a number of regions concerning color, texture, or contour, and extracts for each region a set of features. RBIR approaches (see e.g. [Wan01] or [NRS99]) partition an image into a number of homogeneous regions concerning the extracted color, texture, or contour and extract local features for each region. They barely deploy so-called global features over the entire content.

[1] It works in the same manner with Altavista.

In our approach we also automatically segment an image into its segments or regions concerning color or texture (see section 2). We also determine local features for every region, which can be indexed and which help at the search side. So therefore, our PictureFinder system has a deep impact on this research branch.

### 1.3 Similarity Search

Image similarity search approaches are based for example on a robust method for image similarity by so-called transportations between pixel signatures [HRT01]. A transportation between two signatures is a pixel-oriented operation like translate, import, or align. A similar approach using geometrical transformation can be found in [WGFJ98]. Or a histogram-based method using spatial information about the localization and corresponding color [CCLa01]. This approach indexes images by computing vectors concerning the position and color of pixel belonging to the same region. Spatial information come in PictureFinder also into account, since greater regions with their position can be weighted stronger than small regions. In [Bra00] a probabilistic approach is proposed. It combines the likelihood probabilities of the classes over a framework of different levels and compute the corresponding posterior probability to a certain class. A fractal-based approach can be found in [N.N02].

Again, following the RBIR ideology the basic idea of the PictureFinder system is a similarity search about regions (see section 3). This has the following advantages: first, we are able to take spatial relations of regions into account, second we are able to choose only this regions for the search, which are important in view of the user by weighting for example the size of the region, and therefore third we can limit the search space, because in the frame of search the user can select the regions he wants to search for.

### 1.4 Composition of the Paper

In section 2 we give a brief overview of the analysis methods and therefore of how PictureFinder automatically generates these above-mentioned regions as the basis of RBIR. In section 3 we describe the formulation of a graphical query based on regions and how the search is performed. Additionally, we describe our approach to similarity search, which is in fact a region-based graphical query, too. In section 4 we give some first experimental results. We conclude in section 5 with a brief overview of our ongoing work.

## 2 Overview of the PictureFinder System

PictureFinder is a system for the automatic annotation of large image databases. It contains analysis modules to extract color and texture and a search engine to perform a graphical image search. The analysis results in an automatic generated color and texture index of each image. The index containing color and texture properties is organized within a relational database system.

The PictureFinder search engine allows graphical queries for images containing certain color or texture properties within specified regions. An example of a graphical query for images containing the color yellow in the middle and green in the lower left part of the image is shown in figure 3 on the left. The results of this query are presented in figure 3 on the right.

There is also a web demo of the PictureFinder system which is available under the URL `http://www-agki.tzi.de/bv/pfdemo/`. The image database of the web demo contains approximately 1700 images.

In the following sections we will give a brief overview of the image analysis parts of the PictureFinder system.

## 2.1 Color Analysis

The PictureFinder color analysis is divided into two processing steps: first, the color segmentation of the image in order to obtain homogeneous colored image regions, and second, the indexing of these color regions.

The goal of the color segmentation is to divide an image into different regions which have homogeneous colors. An overview of existing color segmentation methods can be found in [SK94]. Many of them are based on well-known methods used for the segmentation of gray scale images, for instance region growing, edge detection, or cluster analysis.

The PictureFinder color segmentation method is carried out in the HLS color space which uses hue, lightness (brightness), and saturation to define colors[2]. The HLS color space is closer to the human color perception than the more frequently used RGB color space [Fai98]. In a first step, the image is transformed from the RGB to the HLS color space [FvDFH93]. To perform the color segmentation, we use a centroid region-growing approach. The centroid region growing compares each new pixel to be assigned to an average of all pixels already belonging to the region. First, every pixel of the image is processed line by line, from the upper left to the lower right corner, and assigned either to an existing color region or to a new one. That means, the pixels above and left of the current pixel are already assigned to regions. The color distances are calculated between the current pixel and the average color of the up to four already existing regions in the neighborhood. If we find at least one region with a color distance below the segmentation threshold, the pixel is assigned to the region with the minimum color distance. If all color distances exceed the segmentation threshold, a new region is created, containing the current pixel. Second, neighboring regions are merged if the color distance of their average color values

[2]HLS, HSV, and HSB color models based on the same principle. They use all the hue value for the color (angle), the saturation value, which is at last the ratio between the chromatic and the achromatic parts, for the clarity of the color, and the lightness or brightness value for the luminance. Furthermore, all models based on the color disc, too. They only differ essentially in the angles for the color values. E.g. HLS defines blue at $0^o$ and HSV defines red at $0^o$

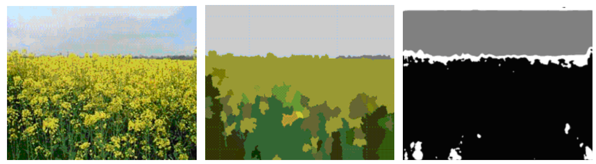

Figure 2: Original image (left), color segmentation (mid), and texture segmentation (right)

does not exceed an allowed threshold. Again the image is processed pixelwise in the same order, but the neighbors considered for merging are the pixels below and right of each pixel. If the neighbor pixels do not belong to the same region as the current pixel and the color distance threshold is not exceeded, the regions are merged. In a last optional step, regions that are smaller than an allowed minimal size are merged with one of their neighbors. This is useful to limit the number of resulting regions. More details are given in [MMH02]. A segmentation example is displayed in figure 2 in the middle.

The result of the color segmentation is a set of color regions, each containing a set of pixels. Simple color features are the average HLS color values of each region. The average HLS color values together with the location of the region build the color index of the image which is stored in the database. The color index is then compared with the graphical search requests.

## 2.2 Texture Analysis

The goal of the PictureFinder texture analysis is the extraction of the texture-based content information. In the same way as the color analysis, it is divided into two steps. In a first step, an image is segmented into regions with homogeneous textures. The texture segmentation is performed by using a bank of linear Gabor filters with n orientations and m frequencies. The input image is separated into $n \times m$ bands, giving the local intensity of the corresponding frequencies [JF91]. Subsequently, a pointwise nonlinear transformation makes all values positive and enhances contrast. The resulting "texture blob" images are locally averaged and then clustered using the Mean Shift algorithm [Che95]. As additional features, the two-dimensional coordinates of each point are used in the clustering process to enhance the spatial coherence of the resulting texture regions. A sample result of the texture-based segmentation is displayed in figure 2 on the right [HM02]. The texture of each region is described using statistical texture features which are mapped to a set of five texture patterns (visual properties), namely homogeneous, speckled, stitched, multiareas, and wavy. The texture patterns together with their location within the image build the texture index of the image which is stored in the database. The texture index is then compared to the graphical search requests.

# 3 PictureFinder Search Functionality

In this section we describe the query functionality of PictureFinder. Based on the analysis methods described above, PictureFinder can search for color or texture or a combination of color and texture attributes. We do not describe the additional query functionality by text. This is a well-known query option.

At first, we describe how a graphical search can be formulated and how this query will be implemented in detail. Subsequently, we describe the implementation of the PictureFinder similarity search. The graphical query is formulated via painting. The user can paint in different ways the regions he is looking for. PictureFinder offers the following painting options: rectangle, ellipse, polygon (lines), and free-hand polygon. Examples are given in figure 4. The similarity query is automatically formulated based on the color segmentation. An example of an automatically generated region-based query can be found in figure 5 in the middle.

## 3.1 Graphical and Similarity Search

Basically, all queries are subdivided into two steps. In the first step all images within a tolerance range will be determined. In a second step, these images will be ranked. In order to perform the ranking, features like color, texture, and density were compared to the query and a match between the range $[0 \dots 1]$ is determined for every single feature, and the overall result is a weighted average of the normalized matches. The weights are reciprocally proportional to corresponding tolerance values preset by the user.

A static grid, which is overlaid over the image and the query, is the underlying basis in order to search for color or texture. During the analysis (see section 2), all colors and corresponding areas are determined for every grid element. Additionally, unless a texture border runs through a grid element, the degree of conformity with the above-mentioned visual texture properties will also be determined. This information for every grid element is used for an efficient indexing.

Therefore, an image accomplishes the query concerning a plotted colored and/or textured region if a minimum of $x$ percent of the area lies in a tolerance range. If the option *anyplace* is activated, the plotted region (query) could be anywhere in the image.

For every query item (a certain colored and/or textured region), a single resulting set returns from the system containing a list of possible matches (images). These sets are then combined with *AND*, *OR* or *soft-AND*. AND means that the image is in every set and the mean is determined. OR means that the image is at least in one set and therefore the maximum is determined. Soft-AND is the default combination and this means that the assessment of the value is computed to the ratio between the sum of the regions depending on the weights multiplied with a score and sum of the regions depending on the weights.

| color/ranges | erl | crl | crr | err |
|---|---|---|---|---|
| red | 305 | 345 | 10 | 25 |
| purple | 260 | 285 | 320 | 350 |
| blue | 145 | 200 | 265 | 290 |
| green | 75 | 105 | 160 | 215 |
| yellow | 35 | 50 | 70 | 113 |
| orange | 8 | 20 | 40 | 55 |

Table 1: The defined borders of the core and the extended ranges over the color disc concerning the CNS. *Erl* stands for extended range left, *crl* stands for core range left, *ccr* stands for core range right, and *err* stands for extended range right, respectively.

### 3.1.1 Comparison of colors

The PictureFinder system compares the colors in the HSB[3] color space. In doing so, the conformity concerning the hue, the saturation, and the brightness are determined separately, and a weighted mean is computed.

Since in the HSB color space the colors are not equidistantly distributed over the color disc, we define a *core range* and an *extended range* concerning the colors and their corresponding angles. In order to clarify the not equidistant distribution of colors over the color disc, we give the following example. The range of yellow colors is very narrow (an angle of nearly 20 degrees), the blue colors in contrast occupy nearly a quarter of the color disc. The weighting function of the PictureFinder system is oriented on the CNS (Color Naming System [BBK82]). For each main color (red, orange, yellow, green, blue, and purple) a core range between the adjacent quarter colors and an extended range between the adjacent three quarter colors in both directions on the disc are defined. For example, the core range concerning blue goes from greenish-blue to purplish-blue, and the extended range goes from bluish-green to bluish-purple, respectively. In table 1 the defined ranges for the colors red, purple, blue, green, yellow, and orange concerning the CNS are listed.

In order to compare two color angles, a target range concerning the first angle (reference angle) is determined. Within this range a linear distance measure is used. The full match of the two angles is rated with 1.0. A match at the border or outside of the range is rated with 0.0. The target range at the reference angle is determined as follows:

1. Is the color inside a core range, the corresponding core and the corresponding extended range is valid.

2. Is the color inside an extended range on the left (without a core range), then the interval [extend left; core right] is valid.

[3]The HSB is equivalent to HSV. Concerning the CNS it is rotated about $120^o$.

3. Is the color inside an extended range on the right (without a core range), then the interval [core left; extended right] is valid.

4. Is the color between two ranges the union of 1. to 3. is valid.

The interval is downsized or enlarged to both sides in dependence of a percentage setting.

#### 3.1.2 Comparison of hue, lightness, and texture

Concerning the set tolerance value, a search interval is determined. The weighting lies in the interval [0...1].

For very unsaturated or very dark colors (maybe already grey levels) a comparison concerning color angles is not meaningful. Therefore, in this case the tolerance values or the weights, respectively, are set to default values.

Five texture classes, more precisely five visual properties, are used in the actual system (see section 2.2). For every element of the static grid (mentioned above), the match of every texture class is determined. The search considers only those images whose grid elements matches nearly 100% minus a set tolerance.

### 3.2 Automatic generation of a query

In order to search for similar images, regions based on the automatic color segmentation (see section 2.1) are determined. A query region is determined for every minimal sized colored region. In addition, if the sum of ratings for a texture class inside the grid elements under a certain region exceeds a certain threshold, a query region concerning this texture is created, too. And the texture class is assigned to this region. Very small regions are searched for anywhere by default since the probability is very low that this region is at the very same place in the searched image.

The generated query regions are combined with *soft-AND* by default. Therefore, the set weights (or tolerance) are critical for the query success. We defined the following three heuristics:

- *Weighting of the region size*: This means the greater a region the more important is this special region for the similarity sensation.

- *Weighting of saturation/lightness*: This means strong saturated and light regions are more weighted than less saturated and dark regions. Is the lightness greater than 0.4 the weight is determined by the means of saturation and lightness. Otherwise, the threshold is only determined by the lightness.

- *Weighting of the position*: This means central regions are more important than regions at the border of the image. The position is determined by the location of the grid elements overlaid by the region.

All these weightings are combined multiplicative.

# 4 Experimental Results

In the following sections we present an experimental evaluation of the PictureFinder query functionality. In section 4.1, the graphical search is evaluated concerning four graphical queries specified using the graphical search interface. In section 4.2 we present the results of a query by example.

## 4.1 Graphical Search

In order to evaluate the results of the graphical search, we choose a set of 220 images which were analyzed by PictureFinder concerning their color and texture properties. Then, four different graphical queries were specified using the graphical search interface of PictureFinder.

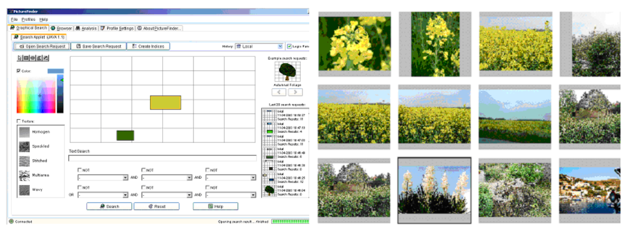

Figure 3: Graphical query with corresponding image results

A human observer had to look at the test image set in order to evaluate manually which subset of the 220 images is expected to be found by each query. Then the queries were submitted to the PictureFinder search engine and the query results were compared to the results expected by the human observer. This leads us to a first idea of precision (percentage of correct results with respect to overall results found by PictureFinder) and recall (percentage of correct results found by PictureFinder with respect to results expected by the user) of the PictureFinder graphical image search.

Figure 4: Evaluated queries

Table 2 summarizes these first results. As an example, the third query (yellow and green region) together with the results obtained from the PictureFinder system is shown in figure 3.

The recall $\mathcal{R}$ is defined by

$$\mathcal{R} = \frac{\gamma \cdot 100}{\alpha}$$

| Query | $\alpha$ | $\beta$ | $\gamma$ | $\mathcal{R}$ | $\mathcal{P}$ |
|---|---|---|---|---|---|
| 1 | 12 | 8 | 7 | 58,3 | 87,5 |
| 2 | 9 | 21 | 8 | 88,9 | 38,1 |
| 3 | 6 | 11 | 6 | 100,0 | 54,5 |
| 4 | 11 | 4 | 3 | 27,2 | 75,0 |
| overall | | | | 68,6 | 63,8 |

Table 2: Query results: $\alpha$ stands for the number of relevant results, $\beta$ stands for the number of results retrieved by PictureFinder, and $\gamma$ stands for the number of correctly retrieved results.

and the precision $\mathcal{P}$ is defined by

$$\mathcal{P} = \frac{\gamma \cdot 100}{\beta}$$

respectively.

### 4.2 Query by example

In addition to the graphical search where the user specifies the query using the graphical search interface of PictureFinder, it is also possible to search for similar images. To do so, the user selects an example image. Then PictureFinder automatically creates a graphical search query based on the results of the color and texture analysis of the example images. An example is shown in figure 5. The generated graphical query may be adapted to the user's requirements, e.g. by deleting regions which do not need to be similar and therefore should not be part of the query.

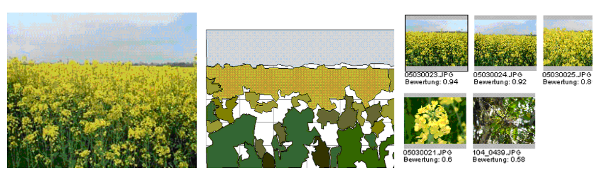


Figure 5: Query by example. Query image (left), corresponding generated graphical query (mid), and corresponding results (right)

## 5 Conclusion and Future Work

In this paper we presented a brief overview of the PictureFinder system in order to generate a text-based index concerning color and texture features appearing in an image and how to perform a graphical search over the image database. We also addressed the PictureFinder similarity approach following the ideas of RBIR, and we presented some experimental results which will be extended. Future work also includes the integration of contour features into the analysis as well as into the search engine. The color search functionality should be extended in order to take the brightness and saturation in combination with the hue value into account.

Furthermore, we are currently developing an automatic object recognition - for a constrained domain - with the help of ontologies. Our idea is not only to use the ontologies at the object recognition state but also at the query state as a thesaurus.

# 6 Acknowledgments

Many thanks to Andree Lüdtke and Hinnerk Becker for their never-tiring work on implementations.

# References

[BBK82] T. Berk, L. Brownston, and A. Kaufman. A new color-naming system for graphics languages. *IEEE CG&A*, 2(3):37–44, 1982.

[Bra00] Ben Bradshaw. Semantic based image retrieval: A probalistic approach. In *Proceedings of the eighth ACM international conference on Multimedia table of contents Marina del Rey*, pages 157–176, 2000.

[CCLa01] L. Cinque, G. Ciocca, S. Levialdi, and A. Pellicano andR.Schettini. Color-based image retrieval using spacial-chromatic histograms. *Image and Vision Computing*, 19:979–986, April 2001.

[Che95] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[Fai98] M.D. Fairchild. *Color Appearance Models*. Addison Wesley Longman, Inc., 1998.

[FvDFH93] J. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics - Pracitce and Principles*. Addison-Wesley Publishing Company, second edition, 1993.

[HM02] Th. Hermes and A. Miene. Domain-independent texture analysis and description. *Kuenstliche Intelligenz*, 16(1):64–69, 2002.

[HRT01] A.S. Holmes, C.J. Rose, and C.J. Taylor. Measuring silmilarity between pixel signatures. *Image and Vision Computing*, 20,Issues 5-6:331–340, April 2001.

[JF91] A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[MMH02] B. Melzer, A. Miene, and Th. Hermes. Bewegungsanalyse in bildfolgen auf der basis eines n:m-matchings von farbregionen. In *Proc. of 8. Workshop Farbbildverarbeitung FarbBV 2002*, 10.-11. Oktober 2002.

[N.N02] N.Papamarkos N.Nikolaou. Color image retrieval using a fractal signature extraction technique. *Engineering Application of Artificial Intelligence*, 15:81–96, 2002.

[NRS99] A. Natsev, R. Rastogi, and K. Shim. WALRUS: A Similarity Retrieval Algorithm for Image Databases. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 395–406, 1999.

[SK94] W. Skarbek and A. Koschan. Colour Image Segmentation – A Survey. Technical Report 94-32, TU Berlin, 1994.

[SWS+00] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[Wan01] J. Z. Wang. SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963, 2001.

[WGFJ98] H. Wang, F. Guo, D. D. Feng, and J. S. Jin. A signature for content-based image retrieval using a geometrical transform. In *Proceedings of the sixth ACM international conference on Multimedia*, pages 229–234. ACM Press, 1998.