A picture is worth a thousand words

Content-based image retrieval techniques

B. Thomée

A picture is worth a thousand words

Content-based image retrieval techniques

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden
op gezag van de Rector Magnificus prof. mr. P. F. van der Heijden,
volgens besluit van het College voor Promoties
te verdedigen op woensdag 3 november 2010
klokke 16.15 uur

door

Bart Thomée
geboren te Delft
in 1981

Promotiecommissie

Promotor:	Prof. dr. J.N. Kok
Copromotor:	Dr. M.S. Lew
Overige leden:	Prof. dr. T.H.W. Bäck
	Prof. dr. A.F. Smeaton (Dublin City University)
	Prof. dr. H.A.G. Wijshoff
	Dr. E.M. Bakker

This research was jointly funded by Leiden University and the Netherlands Organization for Scientific Research (NWO) in the framework of the BSIK/BRICKS project (grant no. 642.066.603) and was carried out in the ASCI graduate school (ASCI dissertation series no. 216).

The cover illustration is Valley Drive by Wolfgang Staudt. This book was printed and bound by Kapsenberg van Waesberge, in association with Gelling Publishing. The paper used is FSC certified.

Contents

1. Introduction

1.1 Finding images of interest

All of us experienced it someday in the past, when we were looking for something particular, which we knew existed someplace, and yet we were unable to find it. Whether it was an old newspaper article cut out years before or a silver necklace that recently went missing, sometimes finding whatever you are looking for can be a lengthy task that requires searching all corners of your home. In the current age with computers capable of performing tasks millions of times faster than any human can, it may come as a surprise that even for a computer it is often not easy to find a particular digital object, especially when it comes to imagery. Of course, a computer is not a super-human, capable of performing all the things we can at a much higher speed. Rather, at the moment a computer is a machine that can only do those things we tell it to do. We tell it what to do by providing it with a sequence of basic instructions that, when executed by the machine, results in the desired outcome. Computers excel at tasks for which there is a clear algorithm, for instance calculating tax returns or solving complex equations. Unfortunately, for other kinds of tasks it is not straightforward to construct such an algorithm. The general problem of image retrieval, the topic we address in this thesis, is one of such tasks.

With the current trend of transferring more and more information to personal computers and the internet, different approaches are required in order to find back the desired information. Search engines like Google and Yahoo! are quite capable of retrieving documents based on their textual contents. However, the quality of the results is often far from optimal when it comes down to finding imagery. The well-known saying "a picture is worth a thousand words" highlights one of the main reasons it is so difficult to track down the images someone is looking for, in particular because the words assigned to an image can also differ from person to person. What one person may describe as a "holiday picture showing a mountain" can be considered by another as "scenery of Iceland", whereas a third person may perceive the photo as "the Eyjafjallajökull volcano on the verge of eruption". A search engine thus has to accommodate for all kinds of image interpretations.

Computers are not yet able to see the world like we do. The field of computer vision aims to translate the knowledge on the human vision system into algorithms to give computers similar capability [Levine 1985]. In this thesis many different computer vision techniques will be discussed, ranging from techniques that focus on the colors of images to techniques that analyze images from the point of view of the way cones in the human retina are distributed. However, in our work computer vision is the means to an end, and we use it in the context of content-based image retrieval. Our main research objectives are to design techniques that (i) assist the user in finding images of interest quicker than before, and (ii) provide the user with a better search experience than before.

1.2 Thesis overview

This thesis is based on first-authored articles that have been published in or are currently under consideration at respected journals and conference proceedings. The research has been carried out during the four-year period of the PhD. The focus of our work has been on developing and analyzing techniques to improve the state of the art in content-based image retrieval. We present work on interactive search (also known as relevance feedback), exploration of image collections, artificial imagination and near-duplicate detection. Because in this digital age the number of pictures on computers, local networks and the internet is already enormous and the speed at which new images appear shows no signs of slowing down, we shift the center of attention in the final chapters to retrieval techniques that are feasible for large quantities of images. We have started development on two internet-based image retrieval systems that incorporate several of our proposed techniques and we present our early work on both systems in the appendices.

Chapter 2: Trends and challenges in relevance feedback-based image retrieval
A survey is presented that reviews over 200 papers published between 2002 and 2010 in the area of interactive image retrieval. The review provides a comprehensive background on the current directions the field is moving towards and also serves to place our work into context. The survey has been submitted to:

- ACM Computing Surveys

Chapter 3: Artificial imagination
We propose a novel retrieval technique that we have called artificial imagination. This technique gives the search engine the ability to 'imagine' by synthesizing images that ideally are similar to what the user is looking for. We present an evolutionary algorithms-inspired method for synthesizing textures and determine whether or not such synthetic images can be beneficial to visual search. This approach has been presented at the following conferences:

- 6th ACM International Conference on Image and Video Retrieval
  Amsterdam, Netherlands, 2007 [Thomee et al. 2007a]
- 2007 IEEE International Workshop on Human-Computer Interaction
  Rio de Janeiro, Brazil, 2007 [Thomee et al. 2007b]
- 19th IEEE International Conference on Pattern Recognition
  Tampa, FL, USA, 2008 [Thomee et al. 2008a]

Chapter 4: Visual exploration and search
One of the grand challenges in our field is the need for experiential exploration systems that allow the user to gain insight into and support exploration of media collections. In this chapter we not only present such an interactive image retrieval system that incorporates a new browsing mechanism called deep exploration, but

also propose an approach for automatic feature weighting. The basis for this chapter is formed by publications in the following conference proceedings:

- 6th IEEE International Symposium on Image and Signal Processing
  Salzburg, Austria, 2009 [Thomee et al. 2009a]
- 17th ACM International Conference on Multimedia
  Beijing, China, 2009 [Thomee et al. 2009b]
- 21st Benelux Conference on Artificial Intelligence
  Eindhoven, Netherlands, 2009 [Thomee et al. 2009c]

Chapter 5: TOPSURF: a visual words toolkit
TOP-SURF is an image descriptor that combines interest points with visual words, resulting in a high performance yet compact descriptor that is designed with a wide range of content-based image retrieval applications in mind. TOP-SURF offers the flexibility to vary the descriptor size and supports very fast image matching. Besides the source code for the visual word extraction and comparisons, we also provide a high level API and very large pre-computed codebooks targeting web image content for both research and teaching purposes. A paper on this descriptor has been accepted for publication in the conference proceedings of:

- 18th ACM International Conference on Multimedia, Open Source Competition
  Firenze, Italy, 2010 [Thomee et al. 2010]

Chapter 6: Near-duplicate image detection
In this chapter we focus on imagery available on the internet and evaluate in which ways near-duplicate images differ from each other. We provide a comparative study of content-based near-duplicate image detection methods and specifically target their performance in relation to their descriptor size, description time and matching time to assess their feasibility of application to large image collections. An early version of this work was presented at:

- 1st ACM International Conference on Multimedia Information Retrieval
  Vancouver, BC, Canada, 2008 [Thomee et al. 2008b]

An improved and extended version has been submitted to:

- ACM Transactions on Information Systems

Appendix A: Noteworthy image search
In this appendix we present an image retrieval system that brings together many of the techniques we propose in this thesis. The search engine aims to provide a personalized search experience and focuses on returning images that are noteworthy to the user. We discuss the design of the search engine in detail and offer insight into techniques for handling large amounts of images.

Appendix B: Touch-up! image search
Taking the artificial imagination approach a step further, we present an image retrieval system that is based on the principles of scene completion. The search engine allows the user to interactively erase unwanted parts of an image and have them replaced by content that she is interested in. The idea is that these touched up images will lead to better retrieval results, because they more closely match what the user is looking for.

The scientific contributions in this thesis are as follows. In Chapter 2, we present the most comprehensive survey to date of relevance feedback-based image retrieval. In Chapter 3, we present one of the first systems which utilizes an artificial imagination in the context of image retrieval. Moreover, we show that synthesized imagery can significantly reduce the required user feedback for finding the desired imagery. In the research community, most methods focus on either exploration/browsing or search, but not both. In Chapter 4, we present a novel search interface which facilitates both image database exploration and search. Currently, there is a need for scalable image retrieval descriptors, that is, descriptors which are effective on very large image databases. In Chapters 5 and 6, we introduce several novel descriptors which have a very low memory cost, yet high accuracy, even for large image collections.

Our research shows high potential to serve as the foundation of future research projects. Currently, our techniques are being combined into a single internet image retrieval system. For instance, a search engine could use the TOP-SURF descriptor to find similar images, while allowing the user to easily browse through the image collection and at the same time asking the user for feedback on artificial images to get a better sense of the user's interests. We have already started working on the noteworthy search engine (Appendix A), which uses the TOP-SURF descriptor, and the touch-up! search engine (Appendix B), which extends the artificial imagination approach.

1.3 Standards, definitions and terminology

For those unfamiliar with the field of image retrieval the following definitions and terminology are worthwhile memorizing.

Category: A group to which images can be assigned that usually have one or more particular characteristics in common.

Class: Same as category.

User: A person that interacts with the retrieval system, where the person can be real (i.e. human) or simulated. Simulated users are often used to get a rough idea of an algorithm's

performance. Note that we use feminine pronouns in this thesis when referring to a user.

Label: A judgment that is assigned to an image, which can for instance be **positive** or **negative**, a particular **category**, etc.

Positive: An indication given by the **user** that the image in question is what she is looking for.

Relevant: Same as **positive**.

Negative: An indication given by the **user** that the image in question is not what she is looking for.

Non-relevant: Same as **negative**.

Irrelevant: Same as **negative**.

Neutral: An indication given by the **user** that the image in question is neither **positive** nor **negative**.

Ground truth: The true assignment of labels to images, which is agreed upon before any **experiments** are performed.

Experiment: The whole process of **training** and **testing** an algorithm.

Training: Preparing an algorithm for **testing** by letting it analyze **training data**.

Testing: Running an algorithm by letting it analyze **test data** in order to determine its **performance**.

Training data: A set of images that are representative for the **test data**, but do not appear in that set.

Test data: A set of images that will be used during **testing**.

Evaluating: Verifying/validating the **performance** of an algorithm by analyzing the results obtained from **testing**.

Performance: Quantitative and/or quantitative assessment of how well an algorithm reaches predefined targets, e.g. the number of images correctly **labeled** as belonging to a particular **class**.

Benchmarking: Comparing the **performance** of an algorithm with that of other algorithms.

Relevance feedback: The **user** voicing her opinion to the search engine about one or more images by indicating their **relevance**.

Annotation: A comment added to an image, usually in the form of a **keyword**, although it can also specify the location of objects, etc.

Keyword: A description consisting of one or more words.

Tag: Same as **keyword**.

In the world of image retrieval there are no official standards, and in the literature we can find many different ways of conducting experiments and evaluating results. Because of the lack of standards it is often very difficult to compare new techniques with existing ones, unless the results are obtained under the exact same conditions, i.e. using at least the same image collections, the same query images and the same performance measures. Still, some practices in our field are so widespread that they can be considered as unwritten rules or guidelines, effectively having become standards. In the last decade several papers have been written on standards in evaluation and benchmarking [Marchand-Maillet and Worring 2006; Huiskes and Lew 2008a] and a number of frameworks have been proposed to facilitate intermethod comparisons [Jin et al. 2006], but so far the status quo remains: our field consists of many little, loosely connected, research islands. On the positive side, there are many promising initiatives, such as the ACM International Conference on Multimedia Retrieval that is slated to be the primary ACM meeting on multimedia retrieval, image retrieval evaluation projects (i.e. ImageCLEF [Müller et al. 2010]) and video retrieval evaluation campaigns (i.e. TRECVid [Smeaton et al. 2006]). In Chapter 2 we discuss the current state of the field of image retrieval in more detail.

1.4 Common choices

To further introduce our research area, we will now look at common choices for image collections, image descriptors, similarity measures and performance measures.

1.4.1 Image collections

We can distinguish between several types of image collections, each used in a different area of image retrieval. Much work mainly focuses on (i) *general images*, (ii) *textures* and (iii) *objects*, although other collections such as face databases and medical imagery are frequently used as well. In our work we have only used imagery from the first two categories.

1.4.1.1 General images

The general images category is the broadest and most widely used in the research literature. In this category images belong that depict just about anything one encounters in everyday life.

Corel

The original Corel stock photo collection consists of over more than 800 CDs, each containing images of a particular category. These categories can be very broad, e.g. ‘ocean’ and ‘germany’, or can be more limited of scope, e.g. ‘sunset’ and ‘religious stained glass’. Some examples of the collection are shown in Figure 1.1. Due to its sheer volume the collection is never used in its entirety, which unfortunately has led to the situation that every research group creates their own Corel subset for use in their experiments, e.g. Corel5k [Duygulu et al. 2002]. Because the complexity of image categories varies from one to another, retrieval systems that use different subsets are difficult to compare directly. Nonetheless, the Corel collection is well-known throughout the research community.

Figure 1.1: Example photos from the Corel collection.

MIR-FLICKR

The most recent substantial additions to the set of available image collections are the MIR-FLICKR 25,000 [Huiskes and Lew 2008b] and MIR-FLICKR 1,000,000 [Huiskes et al. 2010] sets. Both contain images collected from the Flickr photo sharing website and all images are made available under Creative Commons attribution licenses. These licenses are liberal enough to at least allow the use of the images for benchmarking purposes. This is in contrast to many other collections where the images are copyrighted and officially should not be used. All images include the tags that the original photographer has assigned to them. Additionally, in the 25k image set all images have been manually annotated by several annotators, making this one of the largest image collections of its kind. Even though it has only been available to the research community for a short time, the popularity of the MIR-FLICKR sets is rapidly increasing. Some example images are shown in Figure 1.2.

Figure 1.2: Example photos from the MIR-FLICKR 1,000,000 collection, uploaded by the following users (left to right): silkegb, Dia™, Takashy, and Richard_Miles.

Web

With current estimates putting the number of images available on the internet into the tens of billions, the web provides a great source of imagery for our community. Because the images on the internet are of diverse modality, e.g. logos, graphics, celebrity shots and stock photography, it is possible to use them to create quite challenging image collections. To obtain images from the web, researchers generally write a so-called internet crawler that will download all images it encounters on the websites it visits. We have done the same, and our internet crawler is described in Appendix A. Because images on the web come in all shapes and sizes it is common to leave out small icons and banners. In Figure 1.3 we show some example images from the internet.

Figure 1.3: Example images from the internet.

1.4.1.2 Textures

Textures are images that contain a certain kind of pattern, which often have a very uniform (homogeneous) structure, although this is not always the case. Textures are an important part of life, since they often are an intrinsic quality of a particular object or concept, e.g. the fur of bears, the streakiness of grass and the roundness of pebbles.

VisTex

The Vision Texture library [Pickard et al. 1995] was originally created as a freely available alternative to the heavily copyrighted Brodatz collection [Brodatz 1966], where permission to use the images must be explicitly obtained. Besides offering colored homogeneous texture images, the VisTex library also contains real-world scenes that contain multiple texture patterns, see for example the images shown in Figure 1.4. The collection is no longer maintained, but remains available to the public.

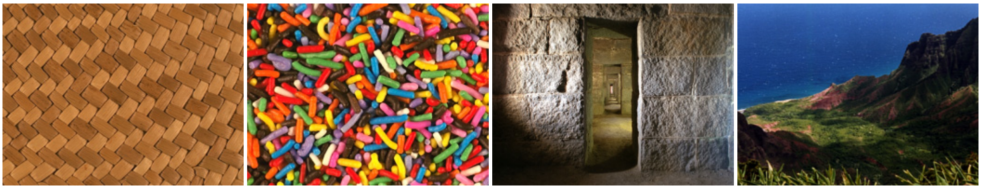


Figure 1.4: Example textures and scenes from the VisTex collection.

Ponce

The Ponce collection [Lazebnik et al. 2005], consisting of 25 categories each containing 40 grayscale images, is frequently used by the texture retrieval community, because it is well-known, easily available, and considered to be challenging. Several textures are shown in Figure 1.5.

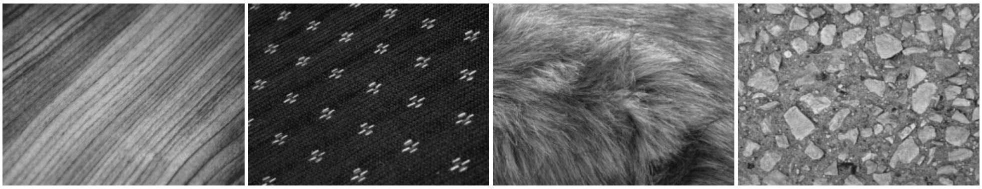


Figure 1.5: Example textures from the Ponce collection.

1.4.1.3 Objects

In order to detect the presence of certain objects in images it is imperative to know beforehand what they look like. Several object databases are available that provide a number of scenes that are all pre-annotated, usually containing the labels of the objects and their locations in the image. Such databases can be used to train an object detector.

Caltech

The Caltech 101 [Fei-Fei et al. 2006] and Caltech 256 [Griffin et al. 2007] datasets, see for example Figure 1.6, are used by many researchers for object recognition. The collections contain 101 and 256 object categories, respectively. As is the case with the Pascal VOC collections, each of the images is accompanied by annotations that indicate the bounding box around the object and its outline.

Figure 1.6: Example objects and scenes from the Caltech 256 collection.

Pascal VOC

The Pascal Visual Object Classes library [Everingham and Winn 2007] is an annually changing library of scenes that is developed for the Pascal Challenge. The dataset is annotated with the ground truth, which includes a bounding box for the objects of interest and segmentation masks, see Figure 1.7.

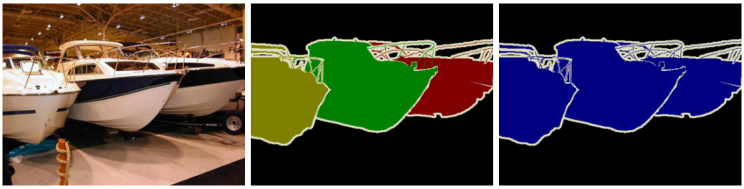


Figure 1.7: Example image from the Pascal VOC collection containing boats (left), the segmentation into individual objects (middle) and classification into the boat category (right).

During the Pascal challenge new (unannotated) scenes are given to research teams, where each team attempts to be the best one at detecting which objects are in the scene and where they are located.

1.4.2 Image descriptors

By itself an image is simply a rectangular grid of colored pixels and to a computer an image doesn't mean anything, unless it is told how to interpret it. Image descriptors are designed for this purpose in the context of image retrieval. Descriptors aim to capture the image characteristics in such a way that it is easy for the retrieval system to determine how similar two images are *from the point of view of the user*. In the following sections we introduce the basic principles of several kinds of image descriptors to give an impression of how images can be converted into a representation that the retrieval system can work with.

1.4.2.1 Colors

A very common and simple way to look at images is by analyzing the colors they contain. For image categories that are tied to a specific color scheme, such as 'forest' and 'sky', color descriptors can yield very good results. For other categories where color plays a smaller role, such as 'car' and 'city', this is not necessarily the case, e.g. a color descriptor is not suitable for recognizing that a yellow Porsche is the same type of car as a blue Porsche. The color histogram has been one of the most popular color descriptors, due to its simplicity and good performance for color-based image categories such as those mentioned above. An example histogram is shown in Figure 1.8. The histogram keeps track of the number of times each color appears in an image. Colors are normally grouped in *bins*, so that every occurrence of a color contributes to the overall score of the bin it belongs to. The bins generally indicate the quantity of red, green and blue found in the pixels, rather than indicating which individual colors are present. Histograms are usually normalized, so that images of different sizes can be fairly compared.

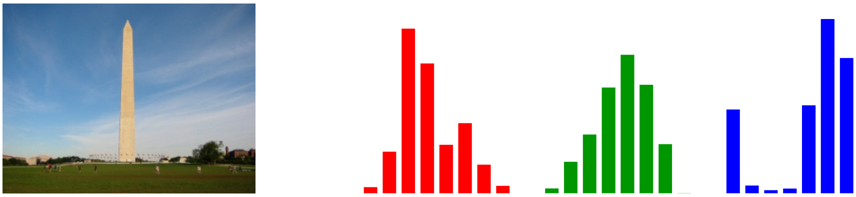


Figure 1.8: Example image (left) and its color histogram (right) with 8 bins per color channel (red, green, blue). Each bin is assigned to a particular part of the color spectrum.

1.4.2.2 Transformations

Transformations can be used to decompose an image into basic (mathematical) building blocks. One popular technique is the Karhunen-Loève transform, which uses a training set of images to discover the *principal components* that account for most of the variability in the data. Each database image can then be described as a linear combination of these principal components. Ideally, similar images have similar coefficients for each of the principal components.

Alternative decomposition techniques are the discrete wavelet transform (DWT) and the discrete cosine transform (DCT), where the images are represented by mathematical basis functions, i.e. wavelets in the case of the DWT and cosine waves in the case of DCT. In Figure 1.9 we show a visualization of the discrete wavelet transform, where an image is decomposed into its low frequency and high frequency wavelet components.

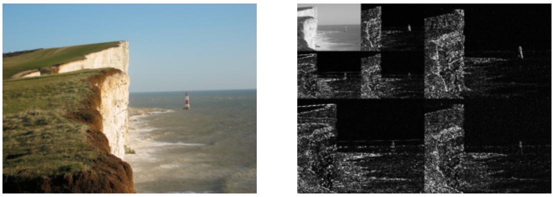


Figure 1.9: Decomposition of an image (left) into its wavelet components (right).

1.4.2.3 Textures

These descriptors are designed for describing the texture patterns present in images. Naturally they are very appropriate for texture images, but they can in principle be applied to all kinds of imagery. An often-used descriptor is the edge histogram, which identifies several kinds of edges in the image and counts how often they occur. The assumption is that similar images will have many similarly oriented edges in common. We show an example edge histogram in Figure 1.10.

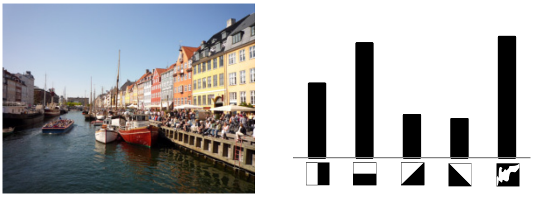


Figure 1.10: Example image (left) and its edge histogram (right), containing the number of vertical, horizontal, 45-degree, 135-degree and non-directional edges.

1.4.2.4 Interest points

An interest point detector analyzes an image with the aim of finding locations that are 'interesting' in a certain way, see for example Figure 1.11. At such locations there are usually sudden significant changes in brightness, coloring or texture. The interest points are designed to be robust to various transformations, such as rotation, scale, translation and illumination conditions. This robustness ensures that images that depict the same scene will still be matched, even if, for instance, they show the scene from different viewpoints. Each interest point is generally described by (i) the location where it was found in the image, (ii) the scale at which the image was inspected when the interest point was detected, and (iii) the orientation of the interest point, referring to the direction in which a change was perceived.

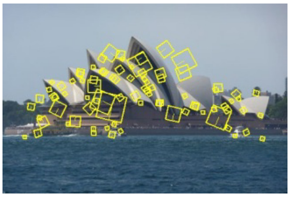


Figure 1.11: Detected interest points in an image, showing their scale and orientation.

1.4.3 Similarity measures

As we will also see in Chapter 2, a large number of different similarity measures are used by the research community. The choice of similarity measure depends on the chosen image descriptor, and may require designing a unique similarity measure if no existing ones are suitable. In this section we will discuss a selection of widely used measures for metric-based and histogram-based image descriptors.

1.4.3.1 Metrics

When the image descriptor consists of a coordinate vector that indicates a point in a multi-dimensional metric space, the similarity between descriptors is commonly determined by calculating the distance between their points in space. Various metrics can be used for this calculation.

Manhattan metric

Also known as the L_1 distance, this similarity metric measures the distance d between two points $x = \{x_1, \cdots, x_n\}$ and $y = \{y_1, \cdots, y_n\}$ as the sum of their absolute coordinate differences:

$$d_{L_1}(x, y) = \sum_{i=1}^{n} |x_i - y_i| \ . \tag{1.1}$$

Other names for this metric are the city block distance and taxicab distance, since they refer to the shortest distance between two points in a city where the streets are laid out in a rectangular grid, such as is the case on Manhattan Island, New York.

Euclidean metric

This metric is commonly referred to as the L_2 distance, and measures the shortest path between the two points:

$$d_{L_2}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}\ . \tag{1.2}$$

When a researcher simply states “the distance between points A and B is X”, without specifying which distance measure is used, the Euclidean distance is generally implied, since it is the most commonly used similarity measure.

Minkowski metric

The Minkowski metric is a generalization of the L_1 and L_2 metrics, where the order parameter p controls how the distance is calculated:

$$d_{L_p}(x,y) = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p}\ . \tag{1.3}$$

Choosing $p = 1$ results in the Manhattan distance, choosing $p = 2$ in the Euclidean distance and choosing $p = \infty$ in the Chebyshev distance. Fractional distances can be obtained by choosing $0 < p < 1$. Note that such distances are not metric because they violate the triangle inequality.

1.4.3.2 Histograms

Histograms are frequently used in image retrieval, for example the color histograms we discussed before. An alternative use of histograms is in the form of probabilistic distributions, where often the likelihood of an image matching the query concept is considered.

Earth mover’s distance

This metric determines the distance between two weighted distributions X and Y as the amount of work it takes to convert the values of the first distribution into those of the second distribution:

$$d_{EMD}(X,Y) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}c_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}}\ . \tag{1.4}$$

Here $X = \{(x_1, w_{x_1}), \cdots, (x_m, w_{x_m})\}$, where x_i is the cluster representative and w_{x_i} is the weight of the i-th cluster. Similarly, $Y = \{(y_1, w_{y_1}), \cdots, (y_n, w_{y_n})\}$, where y_j is the cluster representative and w_{y_j} is the weight of the j-th cluster. Furthermore, c_{ij} is the distance between clusters i and j, and f_{ij} is the optimal flow in converting distribution X to Y.

Kullback-Leibler divergence

This divergence is an asymmetric dissimilarity measure between two probability distributions X and Y. One way of interpreting its functioning is that it measures the added number of bits required for encoding events sampled from X using a code based on Y. For discrete probability distributions $X = \{x_1, \cdots, x_n\}$ and $Y = \{y_1, \cdots, y_n\}$ the distance from X to Y is defined as:

$$d_{KL}(X \| Y) = \sum_{i=1}^{n} X_i log \frac{X_i}{Y_i} \ . \qquad (1.5)$$

Note that the asymmetry of the Kullback-Leibler divergence is easily noticeable, since the distance from distribution X to distribution Y is not necessarily the same as the distance from Y to X.

1.4.4 Performance measures

When evaluating a retrieval system the objective is to get a sense of its performance. This can be in accuracy, e.g. how many mistakes does the retrieval algorithm make, or in computational performance, e.g. how quickly does the system present the results. In this section we will review a number of popular performance measures. In Section 1.4.5 we will present a benchmarking situation in which several performance measures are used and visualized.

1.4.4.1 Accuracy

The main focus of most researchers is on assessing how well their method performs, especially in comparison with the methods of others. In image retrieval the aim for an algorithm is to correctly indicate which class an image belongs to, or at least make sure that the top N images shown to the user are relevant. We can distinguish four cases when an algorithm assigns a label to an image, which are shown in Table 1.1. The most important case is only the *true positive* one, since the user is interested in being shown relevant images On the other hand, because the number of images shown on screen to the user is limited, the *false positive* case influences the number of correctly labeled relevant images that are presented to the user, since one or more of the shown images may actually be incorrectly labeled as relevant.

Table 1.1: Correct and incorrect labeling of an image.

		ground truth	
		class of interest	class not of interest
label given by algorithm	**positive**	true positive	false positive
	negative	false negative	true negative

Precision
This performance measure is used to indicate how exact an algorithm is in returning the relevant images. If we use the terminology of true/false positives/negatives and we assume that the retrieval system only returns us images that it thinks belong to the class of interest, then we can express precision as follows:

$$precision = \frac{|true\ positives|}{|true\ positives| + |false\ positives|}\ . \tag{1.6}$$

However, if we assume that the retrieval system returns us a ranking of images and we only look at a few of them, then the following formula expresses precision:

$$precision = \frac{|true\ positives|}{total\ number\ of\ images\ looked\ at}\ . \tag{1.7}$$

The number of images looked at thus far is commonly referred to as the *scope*. When precision values are compared at a particular scope value, the performance measure is called the *precision rate*, and researchers often specify such a value as for instance *p@20*, which in this case means the precision value when the scope equals 20. When precision values are plotted at multiple scopes this graph is called a *precision-scope* graph. If relevance feedback is used by the retrieval system, then the precision is often plotted against the number of iterations. To create this *precision-iteration* graph the scope is fixed at a certain value, usually the number of images that is shown to the user in a single screen.

Recall
Recall is used to indicate how complete an algorithm is in returning the relevant images, i.e. what percentage of relevant images we have found at this stage:

$$precision = \frac{|true\ positives|}{|true\ positives| + |false\ negatives|}\ . \tag{1.8}$$

Here it does not particularly matter how many images are shown on screen or how many incorrect images are returned, since the recall performance measure only focuses on the number of relevant images that are found thus far. Like with precision, if relevance feedback is used by the retrieval system the recall can also be plotted against the number of iterations as a *recall-iteration* graph.

Precision-Recall
By themselves precision and recall can be quite misleading. For instance, a perfect precision means that so far we have only retrieved relevant images, but it does not specify how many images we have looked at. Similarly, a perfect recall means that we have been shown all relevant images, but it does not specify how many images we actually had to look at in order to find them all. Precision and recall are strongly related and usually by positively increasing one the other is negatively affected, e.g. in a search engine that shows pages of images results the recall will go

up but the precision will simultaneously go down when more and more pages of results are looked at. By plotting the one against the other it is possible to get a more complete picture of the behavior and characteristics of an algorithm.

F_1 measure

The F_1 measure, or score, is a weighted harmonic mean that combines precision and recall into a single value by weighting them equally:

$$F_1 = 2\frac{recall \cdot precision}{recall + precision} \; . \tag{1.9}$$

Differently weighted versions of this measure also exists that give more emphasis on either precision or recall, but these are not as frequently used as the standard F_1 score.

Mean average precision

By averaging the precision values obtained every time a relevant image is encountered you get a good sense of how well a method overall performs:

$$AP = \frac{\sum_{i=1}^{N} precision(i)}{N} \; , \tag{1.10}$$

where $N = |true\,positives| + |false\,negatives|$. By calculating the average precision for multiple queries and averaging all these values a single value, the mean average precision (MAP), is obtained. The MAP value is commonly referred to as being the same as the area under the precision-recall graph.

1.4.4.2 Computational performance

The accuracy of an algorithm is very important, but it is not the only factor that decides how well an algorithm functions. The computational performance is equally as important. If, for example, an algorithm manages to always and only return the images the user is interested in, but it takes an hour to do so, then overall the algorithm is not very effective. Unlike the accuracy, the computational performance can improve over time, e.g. when technological changes result in faster processors, memory and storage media.

Depending on the purpose of the algorithm, often different aspects of computational performance are emphasized. For retrieval algorithms this usually is the time that it takes between receiving the user's query and showing the user the retrieval results. For image descriptors, the focus can be on the time needed to extract the descriptor and the amount of memory that is required to store the descriptor. For an indexing algorithm, the computational performance factor of importance can, for instance, be the number of hard disk accesses necessary to find the most similar nearest neighbors of a query image.

1.4.5 Evaluation and benchmarking

To properly evaluate an interactive retrieval algorithm human users should be involved. The goal of interactive retrieval systems is to find those images that the user is interested in, by engaging in a dialog to find out exactly what the user is looking for. Besides finding the images of interest, the acceptance of a retrieval system also hinges on how satisfied users are with using it. Because finding real users to participate in experiments and processing their search experiences takes quite an effort, most experiments involve users that are simulated. However, human users are subjective in their judgments, because they do not always correctly indicate the relevance or non-relevance of particular images. Therefore simulated users are not real replacements, because they are programmed to not make any labeling mistakes. For example when looking for images of brown bears, simulated users have complete knowledge of the ground truth labeling of the images that are returned by the retrieval system. Thus when a few brown bears are shown, a few black bears, and a few other animals, the simulated user will correctly label the brown bears as positive and all others as negative. A real user, on the other hand, may make the incorrect judgment of labeling a few of the black bears as positive, simply because they belong to the bear category and the user believes that by labeling any bear as positive the system may return more bears, and hopefully more brown ones, or perhaps because the black bears simply look brownish. Good research experiments should thus involve human users.

We now present a hypothetical experiment to give an impression of how several of the aforementioned performance measures are used and visualized. Suppose we are using an image collection that contains 10000 images, divided over 100 categories of 100 images each. Also suppose we are interested in finding as many images of ‘sunsets’ as we can find, and also in finding as many images of ‘cars’ as we can find, where both categories are represented in the collection. We have developed two algorithms, of which the first is based on color and the second is based on shapes, and we want to see which of them is better by comparing their retrieval performance. During each test case, we perform no more than ten iterations of feedback, and the search engine returns us the top 20 matching images. To make it easier, we simulate 100 users to perform the experiments and we average the results. To start the search we let the search engine show a random selection of images, of which exactly one image falls in the category of interest, to allow the algorithms to be compared fairly without being dependent on a particular initial relevant image. Suppose we obtain the results that are shown in Table 1.2.

If we analyze the results, we observe that algorithm I (the one based on color) is more suitable for finding images of sunset, whereas algorithm II (the one based on shapes) is more suitable for finding images of cars. We also notice that algorithm I is better at finding sunsets than algorithm II is at finding cars, while at the same time algorithm I is worse at finding cars than algorithm II is at finding sunsets. This can also be concluded from calculating the mean average precision, which is shown in Table 1.3.

Table 1.2: Average number of relevant image returned for the categories ‘sunset’ and ‘cars’.

			iteration									
			1	2	3	4	5	6	7	8	9	10
algorithm & category	I	sunset	10.2	13.4	16.2	17.8	18.6	19.5	19.8	19.9	20.0	20.0
		cars	2.1	2.7	3.1	3.4	3.8	3.8	3.9	3.9	3.9	3.9
	II	sunset	4.6	5.3	5.4	5.4	5.5	5.5	5.5	5.5	5.5	5.5
		cars	6.2	8.8	10.1	11.3	12.0	12.7	12.7	12.7	12.7	12.7

Table 1.3: Mean average precision for the categories ‘sunset’ and ‘cars’.

algorithm I		algorithm II	
sunset	cars	sunset	cars
0.80	0.16	0.25	0.51

We visualize the results in Figure 1.12 and in Figure 1.13. It is important to realize that the algorithms are limited by the number of images that are returned to the user, which is 20. So even though there are 100 images per category, the recall value can be at most 20% in this experiment.

It is easy to notice the similarities between the precision and recall graphs, since they are based on the same numerator, i.e. the number of relevant images returned, but only differ in the denominator, i.e. the total number of images returned versus the total number of relevant images in the database. It follows naturally that, when the number of images returned equals the total number of relevant images in the database, the precision-iteration and the recall-iteration graphs will be the same.

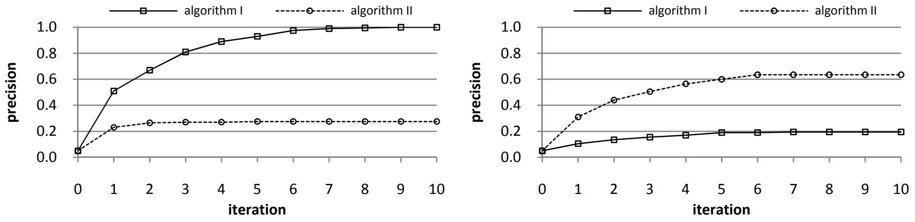


Figure 1.12: Average precision per iteration for the categories ‘sunset’ (left) and ‘cars’ (right).

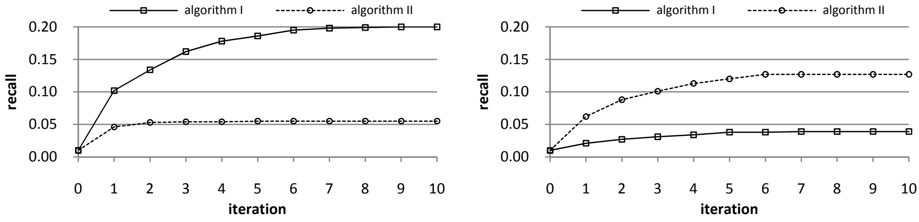


Figure 1.13: Average recall per iteration for the categories ‘sunset’ (left) and ‘cars’ (right).

2. Trends and challenges in relevance feedback-based image retrieval

With the amount of digital information growing at a rapid rate with no end in sight, it is clear that finding an item of interest in this haystack of data will become more and more difficult. In this article we focus on the topic of content-based image retrieval using relevance feedback-based techniques. This survey reviews over 200 articles from recent literature and aims to capture the wide spectrum of paradigms and methods. We also describe several grand challenges for the future.

2.1 Introduction

Many people will look fondly back on an era of old-fashioned analog imagery, in which photographs were in black and white, or perhaps even in color. People will recall the care with which they shot a picture so as not to waste precious film, and they will remember the dark rooms where glossy sheets with scenes and faces emerged. Many of us possess several albums filled with these photos and will occasionally browse through them to relive old memories. Nowadays, with digital technology, the romance of the analog photograph is past, yet on many fronts it has made life much easier. Many more snapshot memories are made with reusable digital storage, and these photos are instantly viewable. It is not uncommon for one's personal computer to contain thousands of photos stored in digital photo albums. At present, billions of images can be found on the internet. One can say the digital age has had a revolutionary effect on how we collect our memories. But with that many images within our reach, how do we go about finding the ones we want to see at a particular moment in time? Looking at each one to find the right image is simply not an option because it is too time consuming.

Content-based image retrieval is the research field that attempts to address this issue of finding the images of interest by analyzing and comparing the content of all images in a collection. Since the early 1990s the field has evolved significantly and has made great leaps forward. “The early years” of image retrieval were summarized in Smeulders et al. [2000], painting a detailed picture of a field in the process of learning how to successfully harness the enormous potential of computer vision and pattern recognition. The number of publications increased dramatically in only a matter of years. The comprehensive reviews of Lew et al. [2006] and Datta et al. [2005, 2008] provide a good insight into the more recent advances in the entire field of multimedia information retrieval and, in particular, content-based image retrieval. In these articles relevance feedback is recognized as one of the most promising topics to further advance the state of the art.

Relevance feedback is an interactive search technique, where the retrieval system engages in a dialogue with the user, with the goal to find out what the user is looking for. The process entails presenting images to the user and soliciting feedback on their relevance over the course of several rounds of interaction, where after each round the system ideally returns images that better correspond to what the user has in mind.

The last review dedicated to relevance feedback in image retrieval was published in 2003 [Zhou and Huang 2003c], but with the rapid progress of technology, many novel and interesting techniques have been introduced since then. To this end, we reviewed all papers in the ACM, IEEE and Springer digital libraries related to relevance feedback in content-based image retrieval over the period of 2002-2010, and selected more than 200 of them for inclusion in this survey. This survey is aimed at content-based image retrieval researchers and intends to provide insight into the trends and diversity of relevance feedback research in image retrieval.

2.2 Relevance feedback from the user's point of view

The relevance feedback process consists of several stages and is shown in Figure 2.1. In the first step, the user issues a query to the retrieval system and shortly after is presented with the initial results. The user can then give feedback on these results in order to obtain improved results. For instance, the user can indicate which images are relevant and which are not, according to what she had in mind. In principle, feedback can be given as many times as the user wants, although generally she will stop giving feedback after a few iterations, either because she is satisfied with the retrieval results, or because the results no longer improve.

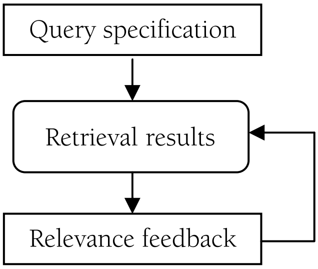


Figure 2.1: Flow-chart diagram of the relevance feedback process.

2.2.1 Query specification

The most common way for a retrieval session to start is with the user providing an example image [Qi and Chang 2007] or by typing in one or more keywords [Kherfi et al. 2004]. The query step can also be skipped directly when the system shows a random selection of images from the database for the user to give feedback on [Thomee et al. 2009b].

In recent literature we find a variety of ways to query the retrieval system when image segmentation is involved. For instance in the work of Amores et al. [2004] the user can draw the outline of the object of interest in the query image and use the object to initiate the search. Alternatively, the user identifies the regions of interest in a pre-segmented query image [Chen et al. 2005; Chiang et al. 2005; Kutics et al. 2003]. Sketching is proposed by Ko and Byun [2002b] and gives the user the opportunity to query the system by drawing one or more search regions and selecting feature constraints, such as whether color is important or not. An interesting way to perform the initial query is presented in Torres et al. [2007], where the user first chooses keywords from a thesaurus that describe the concept of interest, and then selects per keyword one of its associated visual regions.

2.2.2 Retrieval results

The way the results are displayed is most often a ranked list with the database images most similar to the query shown at the top of the list. Because giving feedback on the best matching images does not provide the retrieval system with much additional information other than what it already knows about the user's interest, a second list is also often shown, which contains the images most informative to the system [Huiskes 2006]. These are usually the images that the system is most uncertain about, for instance those that are on or near a hyperplane when using SVM-based retrieval. This principle, called *active learning*, is discussed in more detail in Section 2.3.3.8. Other ways of displaying the retrieval results are discussed in Section 2.2.5.

2.2.3 Relevance feedback

Most retrieval systems give the user the opportunity to give positive feedback only [Jin and French 2003], positive and negative feedback [Zhang and Chen 2005c] or positive, neutral and negative feedback [Yang et al. 2002]. In some of the systems the user can give more accurate feedback: four relevance levels are used in Ko and Byun [2002a] and in Wu et al. [2004a], five levels in Torres et al. [2007] and seven levels in Haas et al. [2004, 2005]. In Huang et al. [2003a] the user can indicate by what percentage a sample image meets the user's initial concepts.

As in the case of query specification, there are several ways to give feedback when the system uses segmented images. Depending on the method of segmentation, Nguyen and Worring [2005] allow the user to give different kinds of feedback. One type of feedback lets the user split or merge image regions, the second type lets the user add or remove detected salient edges in the images, and the last type allows the user to give feedback by drawing a rectangle inside a positive example to select a region of interest. The latter type of feedback is also proposed in Tran et al. [2008].

Besides giving explicit feedback, users can give three types of implicit feedback in Liu et al. [2007a]: (i) Follow Up, indicating that the user likes the results and wants to continue with these images to get better results next time, (ii) Go Back, indicating that the results are worse than the previous iteration, and (iii) Restart, indicating that the user wants to start over again with a different starting point to the query. Implicit feedback is also used in Cheng et al. [2006a, 2006b, 2009] in the form of click-through actions taken by the user on web image search results. Click-through actions refer to the user clicking on an image, for instance with the intention to see it in more detail. These actions implicitly indicate that the user is interested in that particular image, and this implicit information is used to refine the results that are shown to the user in the next result screen. Doing away with the typical keyboard and mouse, Käster et al. [2006] propose to use touch screen gestures and speech recognition. An evaluation on a small user group gave favorable results.

The ostensive relevance feedback model [Campbell 2000] accommodates for changes in the user's information needs as they evolve over time through exposure to new information over the course of a single search session. A dynamically adaptive retrieval approach is proposed, which analyzed the user's browsing history to recommend images she likely would be interested in, with an emphasis on the more recently viewed images over those viewed at an earlier point in time. In a sense, a temporal dimension was added to the notion of relevance. Whereas Campbell only uses textual features for the retrieval of images, Urban et al. [2006b] additionally incorporate visual features. The experiments of Urban et al. demonstrated that users preferred using their ostensive-based retrieval system over a traditional query-based retrieval system.

Kherfi et al. [2002] argue that even though negative images play an important role in relevance feedback, care must be taken on when to use them and how to interpret their meaning. In their system, users can only give positive feedback in the first iteration to allow the system to determine all the characteristic features that every image must possess. In the second step refinement is performed: (i) the desired features should receive more attention, since they only appear in positive feedback, (ii) undesired features should also receive attention since they only appear in negative feedback and thus provide useful discriminating criteria, and (iii) common features appearing in both positive and negative images should be ignored.

2.2.4 Speeding up retrieval

Jarrah and Guan [2008] propose a distributed search system, which aims at improving scalability, availability and efficiency. When the user issues a query, it is sent to all available databases in the user's neighborhood. Each database returns the best results based on its limited image collection using an automatic relevance feedback approach. The search results of all databases are aggregated and returned to the user, after which the user can interactively search on her local machine without needing to issue new requests to the servers. Picard et al. [2008] also propose a distributed system, where the image collections are spread out over multiple hosts. In this system, mobile agents are deployed to visit the hosts, and upon arrival perform a local search. This significantly speeds up the search due to the parallelization of the retrieval process. Over time, the hosts that generally offer more relevant images for a particular query category will be favored over other hosts that do not. Interestingly, their experimental results showed that their distributed system could retrieve significantly more relevant images than a centralized system (that contained all image collections), although it is not clear why this is the case.

Another approach for reducing latency to speed up retrieval performance is presented in Yoon and Jayant [2002], where a pre-fetching mechanism is integrated into the relevance feedback algorithm. The idea is that data that is likely to be requested by the user should be pre-fetched so it is available immediately. This requires that the set of images to be displayed to the user during the next iteration needs to be predicted. This is done by using previous image similarity and relevance feedback information, but depending on the user's need and available network resources, a tradeoff can be made between showing the correct retrieval images (discarding pre-fetched images if they are not necessary after all) or using all pre-fetched images even if not all of them are correct.

By exploiting the characteristics of hard disks, the cluster-based indexing structure of Ramaswamy and Rose [2009] stores elements within the same cluster contiguously, so they can be read via sequential reading operations. The results of their experiments demonstrated that their indexing structure, compared to vector approximation files, required substantially fewer IO reads to find the nearest neighbors of the query image, resulting in a faster search response time. Sequential storage of clusters is also used by Goh et al. [2002] to maximize IO efficiency.

2.2.5 The interface

The role of the interface in the search process is often limited to displaying a small set of search results that are arranged in a grid, where the user can refine the query by indicating the relevance of each individual image. In recent literature several interesting interfaces break with this convention to offer an improved search experience. These interfaces mainly focus on one or a combination of the following three aspects: (i) supporting easy browsing of the image collection, (ii) better presentation of the search results, and (iii) allowing the user to query by grouping images and/or moving images around.

In Fan et al. [2008] a concept ontology is visualized in a hyperbolic way (see Figure 2.2a), allowing the user to obtain an overview of the image collection at a concept level and to interactively navigate the concept ontology by zooming in on different concepts of interest. The retrieval results are visually organized in a cloud, which users can navigate through or click on relevant images to obtain additional images. The interface of Ren and Calic [2009] also uses a hierarchical image representation, where the currently selected image is shown in the center surrounded by the retrieved images, which appear smaller the further they are away from the center image. By clicking on one of the surrounding images, the central focus changes to the selected image.

The notion of visual islands is introduced by Zavesky et al. [2008] to fulfill the principal goal of guided user browsing. This includes a process called island hopping that is used to dynamically reorganize the displayed pages according to the user's selection, so that the user can explore deeper into a particular dimension she is interested in (see Figure 2.2b). Their user experiments showed that users were able to more quickly indicate relevance due to the improved layout, as compared with a regular ranking layout. Thomee et al. [2009a] proposed an interface that allows the user to explore the local neighborhood around images and focus the search on those regions in relevant feature space. This interface is expanded with a technique called deep exploration [Thomee et al. 2009b], which lets the user easily navigate to other promising areas in feature space. This is particularly useful when the search no longer improves with the current set of relevant images.

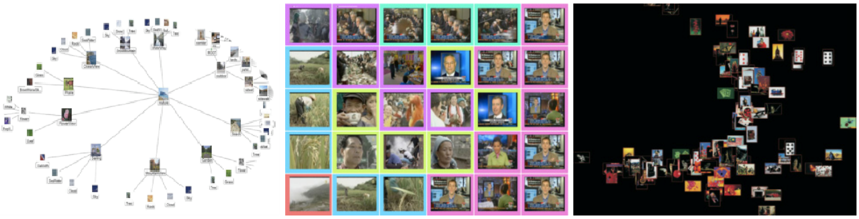


Figure 2.2: Illustration of several user interfaces: a) hyperbolic visualization of Fan et al. [2008], b) visual islands of Zavesky et al. [2008], c) similarity-based visualization of Nguyen and Worring [2008].

Mavandadi et al. [2006] introduced an interface based on data communication theory, from the point of view that there is limited total bandwidth available for transmission, while there are multiple transmitters present that all compete for this bandwidth. The authors argue that an equal division is often not the optimal strategy, and it is usually better to allocate each transmitter a slice of bandwidth according to its transmission probability. Their approach is hence that more 'bandwidth' will be given to images that are likely to be more relevant to the query than to less relevant images. This notion of bandwidth translates to scaling the area of each image displayed in the interface relative to the probability that the user is interested in that specific image. The interface of Heesch and Rüger [2003] places retrieved images on the screen with the distances to the center representing their dissimilarities to the query image. The user gives feedback by moving the images around: moving it closer to the center indicates the image is more relevant to the query, moving it further away means the image is less relevant. Kozma et al. [2009] use an interface that contains several rings of images, with the outermost rings showing large images and the innermost rings small images. Gaze tracking is considered implicit feedback to estimate the relevance of the shown images. Zooming in is considered explicit feedback, and causes the outer rings to disappear from view and the inner rings to come closer. New rings that consist of freshly retrieved images are then placed at the now available innermost positions.

The principle of query-by-example is extended by Nakazato and Huang [2002] to become query-by-groups, where a group of images is considered the basic unit of the query as opposed to individual images. The interface allows the user to drag positive and negative images to a panel and enclose them by drawing surrounding boxes, to indicate if the group is relevant, irrelevant or neutral. Similarly, in Urban and Jose [2007] the search results are shown in a panel from which users can drag images onto a workspace, which serves as an organization tool to construct groupings of images. For each query issued the groups are ranked in order of similarity, with the most similar ones first. Their extensive user experiments indicated that even though their interface was found to be more intuitive and stimulating than a classic interface, it was also more difficult to use. Yet, overall the user interface was perceived as being more effective. Nguyen and Worring [2008] suggest a way for the user to select several images at once by dragging a rectangle around images, rather than having to click on each individual image for labeling. This is performed in their interface for similarity-based visualization (see Figure 2.2c), in which three (conflicting) cost functions are optimized for the optimal representation of the collection to the user: (i) the structure preservation cost, which when minimized optimally preserves the relations between images in visualization space, so that similar images tend to be grouped together, (ii) visibility cost that involves the amount of overlap between images, and (iii) overview cost that controls how well the set of one or more representative images from each cluster is representative for the whole collection. A similar interface is presented in Wang et al. [2009], which reduces overlap between images by further spreading them out so they use all available display space.

A way to indicate which areas of an image are relevant is proposed in Guan and Qiu [2007a]. The user is allowed to 'scribble' on images to make it clear to the retrieval system which parts of an image should be considered foreground and which parts background.

2.2.6 Trends and advances

During the last decade we have seen the interface transition from having only a supportive role to playing a more substantial and important role in finding images. No longer is the interface solely used to display a static set of images together with options for indicating their relevance. Rather, many recently proposed interfaces are truly interactive, offering new ways to initiate the search, give feedback and visualize the retrieval results. Furthermore, the increasing popularity of using higher-level descriptors for image retrieval has expressed itself in interfaces that are tailored to support those ways of searching, e.g. the ability to select a region of interest instead of having to select an image in its entirety.

Even though most research is still directed at improving or designing new classification and indexing techniques to reduce the response time to a query, we have noticed an increase in attention to find alternative ways, such as the utilization of implicit feedback and the development of distributed retrieval systems.

2.3 Relevance feedback from the system's point of view

Even though a user is not particularly concerned with the internals of the retrieval engine, they have a large impact on her perception of the system as a whole. Even if the interface is a joy to use, when the set of images returned by the system are not close to what the user is looking for, she will not be pleased. A global overview of a retrieval system is shown in Figure 2.3. The images in the database are converted into a particular image representation, such as a collection of texture features or regions. These representations can optionally be stored in an indexing structure to speed up the search. Once a query is retrieved, the system applies an algorithm to learn what kind of images the user is interested in, after which the database images are ranked and shown to the user with the best matches first. In this section we cover the recent advances on each of these parts of the retrieval system.

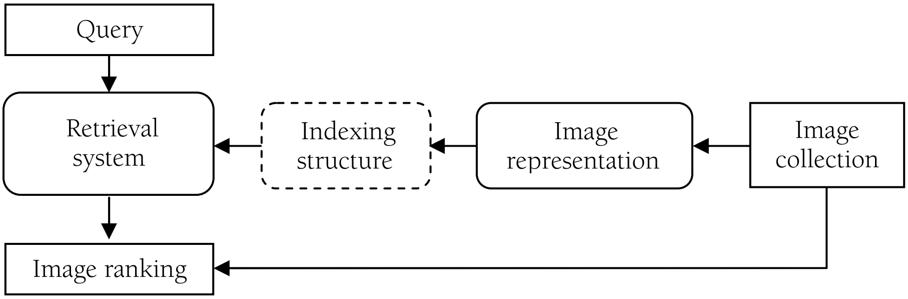


Figure 2.3: Flow-chart diagram of the retrieval process.

2.3.1 Image representation

By itself an image is simply a rectangular grid of colored pixels. In the brain of a human observer these pixels form meanings based on the person's memories and experiences, expressing itself in a near-instantaneous recognition of objects, events and locations. To a computer an image does not mean anything, unless it is told how to interpret it. The future of retrieval envisions systems that will somehow obtain the human experience necessary to perform retrieval tasks accurately and instantly, but for now such experience is provided by researchers through programming. Often images are converted into low-level features, which ideally capture the image characteristics in such a way that it is easy for the retrieval system to determine how similar two images are as perceived by the user. In current research the attention is shifting to mid-level image representations, which focus more on particular parts of the image that are important, such as sub-images and regions, and also to high-level representations, which are designed with semantics in mind, such as concepts and keywords.

2.3.1.1 Sub-images, regions and salient details

Over the past few years, local query-based retrieval has received much attention. In contrast with the majority of relevance feedback approaches, which focus on the image as a whole, local query-based retrieval considers images to be a collection of segments, regions or objects, and assumes the user is only interested in one or at most a few of them.

Most approaches decompose the image into a set of regions using a variety of segmentation algorithms, which commonly are based on k-means [Sun and Ozawa 2005] and mean-shift [Wu et al. 2006], but also for instance on genetic algorithms [Liu et al. 2006b], watershed [Chiang et al. 2005], spectral clustering [Jiang et al. 2005] and max-flow/min-cut [Guan and Qiu 2007b]. Other approaches cut up the image into overlapping or non-overlapping tiles [Shyu et al. 2003] or focus on salient image properties [Nguyen and Worring 2005; Ko et al. 2004].

After the segments, regions or objects have been determined, they are often seen as standalone entities during the search. However, some approaches represent an image in a hierarchical or graph-based structure and exploit this structure when searching for improved retrieval results. In Li and Hsu [2008], images are represented by attributed graphs, with regions as nodes and region connectivity as edges. This turns the region correspondence problem into an attributed graph matching problem, where relevance feedback updates the 'ideal data graph'. Fan et al. [2008] use a four-layer hierarchy, where the low-level features are connected via salient objects to atomic image concepts all the way up to high-level image concepts. A two-layer self-organizing map is used in Chow et al. [2006], where each database image is associated with an upper layer neuron and each region is associated with a lower layer neuron. Image retrieval is done by finding the best matching bottom layer neurons first and then using these as input for finding matching images from the upper layer. A self-organizing map is also used in Zhang and Zhang [2004a] to map similar region features together while separating different ones apart. Luo and Nascimento [2004] hierarchically partition images into overlapping tiles and store them in a tree. Searching for matches is done by 'floating' the tree of the query image over the trees of the database images at different scale levels. A tile-reweighting scheme is used when the user gives feedback, which gives penalties to positive tiles if they are too similar to negative tiles. In the experiments that were performed using existing algorithms as references we noticed that both the graph matching technique of Li and Hsu and the self-organizing map of Chow et al. were shown to outperform the well-known integrated region matching method of Li et al. [2000], whereas the self-organizing map of Zhang and Zhang demonstrated better results than the region-based unified feature matching technique of Chen and Wang [2002]. Note that the latter authors had already shown in their work that their unified feature matching technique was an improvement on integrated region matching. It thus appears that the integrated region matching technique has been superseded by the more recently proposed unified feature matching, graph matching and self-organizing map techniques.

The multiple instance learning and bagging approach lends itself very well to region-based image retrieval, because an image can be seen as a bag of regions. Bag-of-regions is more frequently referred to as bag-of-words, since it originates from text retrieval. In the original bag-of-words model, a textual document is represented as an unordered collection of words, i.e. the representation of a document can specify that it contains a particular word, but it does not say where exactly and it may or may not indicate how many times the word appears. Similarity between documents can be determined by comparing the words (terms) both contain, possibly weighted by a measure of how common the words are. One popular way of term weighting is to use a technique that is based on tf-idf [Salton and McGill 1983]. When translating this technique to images, an image can be represented as an unordered collection of *visual words*, where these visual words can for instance be regions, patches or objects. Similarity between visual bags can be determined by using an approach similar to one where textual bags are used. By incorporating relevance feedback, the idea is that the user can only give feedback on the entire bag (i.e. the image), although she might only be interested in one or more specific instances (i.e. visual words) in that bag. The goal is then for the system to obtain a hypothesis from the feedback images that predicts which visual words the user is looking for [Chen et al. 2005; Zhang et al. 2005b; Huang et al. 2003a, 2003b; Tran et al. 2008]. In Rahmani et al. [2005, 2008] multiple hypotheses are generated starting with randomly selected sets of positive bags, but with different scale factors for the weighting given to regions. The hypotheses are then combined to obtain the image ranking. A prototype-based approach is proposed in Fu and Robles-Kelly [2009], where from each positive bag that represents a relevant image the least negative instance is selected as its representative prototype. This prototype is determined by modeling the distribution of the negative instances and using the distribution to guide the selection. The prototypes are then used in the construction of an SVM-based classifier. Chen et al. [2006] break with the convention of how bags and instances are used, because their multiple instance learning technique does not assume that a bag is positive when at least one of its instances is positive. Rather, they create a feature space from the training instances and new bag instances are mapped into that space. In this space any learning technique can then be applied.

2.3.1.2 Semantics: concepts

The way semantics are expressed is usually in the form of concepts. Concepts are commonly seen as a coherent collection of image patches (‘visual concepts’) or sometimes as the equivalent of keywords (‘textual concepts’). In this section we will focus on visual concepts only.

A fixed number of semantic concepts is discovered or chosen beforehand

Chatzis et al. [2007] use a representative training set for each semantic concept (e.g. ‘birds’, ‘cars’), and find the best fitting *t*-distributed mixture model through the use of an expectation-maximization (EM) algorithm. After the user provides a query image, the best matching semantic class is determined. Any further positive relevance feedback given by the user adapts the corresponding mixture model by establishing the user's target distribution of the semantic class and then adjusting the class model parameters. The assumption of Dong and Bhanu [2003a] is that the database image distribution in feature space is a Gaussian mixture, where each component in the mixture represents an image concept. The number of concepts is discovered by running an EM algorithm on the database. In Zhang and Zhang [2004a] a visual dictionary containing code words is used to describe each image. Similar to the previous two works, an EM algorithm is used to discover the hidden concepts from the visual dictionary. Each image can then be expressed by both code words and concept probabilities. During relevance feedback, the query point in 'code word space' is moved toward good points and away from bad points and is expanded with extra code words that belong to selected relevant images. The representation of the query point in ‘concept space’ is then used to determine similarities with other images in the database. A semantic support region for each concept is learned a priori in Lim and Jin [2005] and each image is expressed as a combination of such regions. During queries, the correspondence between semantic support regions is determined to obtain the ranking.

The number of semantic concepts is discovered automatically

In contrast with the predetermined number of concepts in Dong and Bhanu [2003a], an adaptive model selection is implemented in Dong and Bhanu [2003b], where a number of EM algorithms is applied to models with varying numbers of concepts. The model that is most consistent over time is used, although less consistent models can sometimes also be used to ensure exploration of the search space. In their experiments they demonstrated that, after more and more retrievals were performed, their model was better at correctly identifying the true number of concepts in the database. In Lu et al. [2006] each image region stands for an image concept, and weighted semantic relationships can be established between images based on their regions. Relevance feedback causes the semantic relationships to be modified, so that over time the existing hidden semantic concepts in the database become apparent. A similar approach can be found in Fung and Chung [2007], where a cluster of common visual information is determined after each retrieval session. This cluster can be merged with one or more existing clusters if they are similar enough in a visual or semantic way.

2.3.1.3 Semantics: context

A different view of semantics is proposed in Bartolini [2006], where a query image can be complemented by a context model. This model is a set of significant images (possibly not relevant to the query) that describe the semantic meaning the user is interested in. After feedback, the set of complementing images is automatically adjusted if the system finds that the context has changed. By learning from user's search sessions a mapping is created for future sessions, where the initial image query and used image context are associated with an optimal starting query point and an optimal set of weights.

2.3.1.4 Semantics: keywords

Even though the saying *a picture is worth a thousand words* can be regarded as the tagline of the content-based image retrieval research community, it is generally accepted that the current state of the art image analysis techniques are not able to capture all meanings that an image may have. To enhance the retrieval process, it is worthwhile to combine image features with other sources of knowledge. Low-level visual features alone do not completely convey the content of an image, and text annotations by themselves are generally directly related to the high-level semantics of an image. Together, visual features and annotations can complement each other to provide more accurate results.

A thesaurus, such as WordNet [Fellbaum 1998], is often used to link annotations to image concepts. In Ferecatu et al. [2008] a set of core concepts is identified and the image annotations are linked through synonyms, hypernyms, hyponyms, etc. with these core concepts. Kutics et al. [2003] first map the low-level features of salient image regions to simple keywords and then map them using manual annotation or WordNet to contextual keywords. The keywords are weighted to express their relevance to objects in an image, and are updated each time relevance feedback is given. A semantic hierarchy is built using WordNet in Yang et al. [2005] to explore the relationships between keywords. In Zhang et al. [2003] a two-layer semantic network is created, where at the top layer images are linked to one or more keywords, and at the bottom layer keywords are linked to other keywords through the use of a thesaurus. The weights on the links between images and keywords are updated by user feedback. A similar approach can be found in Lu et al. [2003], although the keywords in the semantic network are not linked through a thesaurus but rather through initial annotation. During a keyword search in both the methods of Zhang et al. and Lu et al. the query is expanded to also include the features of all database images that are associated with the supplied keywords. Ferecatu et al., Kutics et al. and Lu et al. all demonstrated in their experiments that the retrieval results improved by integrating keywords into the search process.

The option to search using keywords is a very convenient way to start a new retrieval session and to prevent many unwanted images from showing up in the results. It is not hard to imagine that manually annotating large collections of images is a very tedious and subjective task, and there is a risk of inconsistent term assignment unless a fixed set of terms is used. Therefore more and more research is directed at automatic image annotation. Although the performance of the current techniques is not completely satisfactory yet, the results are promising and the quality of the annotations is likely to greatly improve over the next years. Using relevance feedback to drive automatic image annotation is still a young field at the moment. One example is described in Zhang et al. [2003], where a Bayesian learning approach is used to propagate the common keywords found in positive feedback examples to the images that have high probability of belonging to the semantic class that the keywords represent. However, most research on automatic image annotation focuses on performing the annotations offline. An interesting approach is proposed by Tsikrika et al. [2009], where click-through data is collected from search logs as a source of concept training data for aiding the automatic annotation algorithm. In Yang et al. [2005] manually annotated images are clustered through k-means clustering on their low-level features and a statistical keyword selection algorithm assigns keywords to the clusters. The unannotated database images are then annotated through content analysis by looking at their probabilities of belonging to one or more of these clusters. The algorithm used in Lu et al. [2003] assigns keywords based on the comparison between the low-level features of unannotated images with the average feature vectors of pre-defined image categories. The top few keywords associated with the most probable matching category are selected. Fan et al. [2008] use 'mixture-experts' that are specialized in different contextual relationships between atomic image concepts and relevant salient objects. For a given input image the underlying salient objects are automatically detected and their features extracted, after which the co-appearance pattern of the salient objects is determined. Bayes' rule is then used to classify the image into the most relevant image concepts and automatically assigns keywords. Liu et al. [2009] reassign the labels, initially assigned to images as a whole, to their appropriate image regions. This is achieved by analyzing all database images and forming semantic regions that are constructed from image patches that are located within similar image regions. These cross-image patch-to-region correspondences are used for the final label-to-region assignment. The keyword propagation scheme of Lu et al. [2009] assigns keywords to images with a restriction to only absorb the keywords and their confidence scores from its nearest neighbors in feature space. Confidence factors are also used in Ji and Yao [2007], where a keyword dictionary is created by manually labeling the regions of training images. Keyword classifiers are then built and applied to all images in the database for automatic annotation with confidence factors. In their experiments they evaluated their visual and textual fusion-based retrieval algorithm against several other techniques. They showed that their fusion strategy largely improved upon the techniques of Rui et al. [1998] and Tong and Chang [2001], and slightly improved upon the more advanced techniques proposed by Xie and Ortega [2004] and Tao et al. [2006b].

One of the newer topics in image retrieval is finding the best balance between using keywords for searching and using visual features for searching. In Cheng et al. [2006a, 2006b, 2009] the textual and visual features are initially kept separate. Thus when analyzing feedback, an optimal query feature vector is obtained in textual space and another optimal query feature vector in visual space. The final image ranking presented to the user is composed first by using the textual query vector to rank all database images, and then using the visual query vector to re-rank them. Wang et al. [2005c] use web images and describe each image by three attributes: hyperlinks, surrounding text and low-level content features. Each of these attributes is modeled in a separate graph, with weights describing the similarities between pairs of images for that particular attribute. After providing a query image, a narrowed-down subset of data is extracted from each of the graphs, which are fused into a new graph. This new graph is then used to train a graph-based support vector machine. The images with the highest positive scores are returned to the user as a ranking, and the images with the highest information scores are returned for further labeling. A graph-based approach is also used by Urban and Jose [2006a], where images and all their features are represented in a multi-layered graph, with one layer for all images in the collection and one layer per feature. The layers contain both visual and textual features. Two types of edges are used, with one representing the relationships between an image and its features, and the second representing the similarity between features, similarity between images and similarity between keywords. During a single search session, the weights between image and feature nodes are adjusted, whereas feedback over time updates the weights on the edges between images when pairs are marked as positive or negative.

2.3.2 Indexing and filtering

Finding images that have high similarity with a query image often requires the entire database to be traversed for one-on-one comparisons. When dealing with large image collections this becomes prohibitive due to the amount of time the traversal takes. In the last few decades various indexing and filtering schemes have been proposed to reduce the number of database images to look at, thus improving the responsiveness of the system as perceived by the user. A good theoretical overview of indexing structures that can be used to index high-dimensional spaces is given by Böhm et al. [2001].

The majority of recent research in this direction focuses on the clustering of images, so that reduction of the number of images to consider is then a matter of finding out which cluster(s) the query image belongs to. In Chiang et al. [2005] c-means clustering is performed using all detected image regions, and during a search session the only images considered are the ones that fall within the clusters where the query regions belong to, or within any nearby clusters. Goh et al. [2002] create clusters by running a pair-wire distance-based clustering algorithm on the database. Each cluster is represented by a subset of images that fall within that cluster. This subset is used for distance calculations to the query image, and the clusters that are closest are then inspected to find its most similar images. Their experimental results indicate that their indexing technique is fast and shows promise to scale up to large datasets. Zhou et al. [2003a] keep track of feedback records, which contain the user relevance evaluations accumulated over time. These records are grouped by performing fuzzy c-means clustering on all image feature vectors. During a new retrieval session, the best matching cluster is identified and the search space is reduced to only use the image contained within that cluster. In Su et al. [2006] clusters are not created beforehand, but during retrieval sessions. After each feedback step, the positive image regions that are similar to each other are first merged via k-means clustering to avoid slowing down retrieval speed, after which group biased discriminant analysis is used to re-cluster each positive class, while scattering negative examples away.

Often the image clusters are stored in a hierarchical indexing structure to allow for a step-wise refinement of the number of images to consider. Wang et al. [2003a] are inspired by the human vision system and use adaptive filter-based feedback technologies to simulate the visual perception model. Due to the non-linearity of the visual perception model, a divide-and-conquer approach is applied to divide the complex non-linear feature space into simpler linear similarity models around the clusters of interest in feature sub-space. A tree model is proposed that has a hierarchical Boolean representation of clustering patters of all feedback samples. In effect, it decomposes a user's complex query concept into a Boolean combination of multiple simpler sub-concepts that span a smaller feature sub-space, which is approximated by adaptive filtering. During retrieval, irrelevant images can be filtered out at each stage of the tree, so they do not have to be considered any longer. In Zhang and Zhang [2005a] k-means clustering is recursively run on all feature vectors corresponding to each region of each image to form a hierarchical indexing structure, where nodes represent centroid feature vectors of their corresponding sets of regions, and the leafs represent a set of regions, where each region points to a set of images that share this region in feature space. Given a query region, the search algorithm is guaranteed to select the cluster whose centroid has the minimum distance in the set of visited nodes. One of the relevance feedback methods employed is to discover the common regions present in the relevant images and those present in the irrelevant images. In comparison with linear search, their indexing technique resulted in a retrieval speedup of approximately four times and only required the inspection of 10-25% of the images, with larger databases needing a lower percentage of inspection. Harnsomburana and Shyu [2002] propose a hybrid search tree for indexing, which consists of an SKD-tree on top and a set of Metric trees connecting to its leaf nodes. The hybrid tree utilizes statistical properties of database images to partition the high-dimensional feature space and stores clusters of images in its leaf nodes, as opposed to individual images. For an image query, the top tree is first searched to locate a set of candidate nodes, which are merged and filtered using a k-nearest neighbor search. The user's feedback causes the results to be split into a relevant and irrelevant set, after which the feature weights are adjusted and the hybrid tree is partially rebuilt. In the experiments their hybrid tree led to a nine time reduction of the number of images to inspect in comparison with linear search. In [Liu et al. 2006b] a genetic algorithm is used to cluster detected image regions. Per cluster all regions are analyzed by a max-flow/min-cut (graph cut) algorithm and outliers are removed. The cluster relationships are modeled into a kd-tree, which gives easy access to the clusters of images that are related to the query region.

Another way to reduce the number of images to consider during retrieval is by partitioning the feature space and only looking at that area of space which the query image belongs to. In Cha [2003] the partitioning is done in combination with clustering. Each feature dimension is divided into a number of intervals, where each interval represents a cluster that contains a fraction of the images in the database. Per dimension cluster a binary bitmap is created that indicates which images are contained within that cluster. The bitmap facilitates finding images within a small range around the query, providing a fast search for k-nearest neighbors in high-dimensional, and supports complex similarity queries with relevance feedback. Tandon et al. [2008] use a B+ tree for each feature dimension. After each round of feedback the optimal feature weights are determined, which indicate the most important features. To reduce the search space only the trees of the most important dimensions are consulted. In their experiments they showed that the average response time to a query was not affected even when more and more images were inserted into the indexing structure. In Yu et al. [2007] the feature space dimensions are partitioned into a number of disjoint subsets of equal size. By exploiting the properties of the Euclidean distance measure and having a bound on the search range, they ensure that no false dismissal will occur when composing the retrieval result image set. Ashwin et al. [2002] enclose the relevant images in a region in feature space, with a decision boundary that is composed of hyperplanes that separate the relevant region from each negative feedback image.

The database images within the relevant region are then ranked on their distance from the separating hyperplanes.

Hashing is a form of space partitioning and is considered to be an efficient approach for indexing. One of these techniques, locality sensitive hashing (LSH) [Indyk and Motwani 1998], has inspired many researchers and is frequently used as a reference when comparing experimental results. In Kuo et al. [2009] a query expansion technique is proposed for locality sensitive hashing so additional buckets are discovered that contain similar images to the query image, which otherwise would not have been found. Two approaches are presented: intra-expansion identifies features that are similar to query features by inspecting neighboring buckets, whereas inter-query expansion obtains new features that are not present in the query image but are present in highly similar images. The latter type of expansion is performed via pseudo-relevance feedback, where new queries are automatically issued using these highly similar images after which the results combined. A different hashing approach is proposed by Yang et al. [2008], which they call randomized sub-vectors hashing. This technique considers feature vectors to be similar when the L2 norms of their randomized sub-vectors are approximately the same, resulting in fast and efficient indexing. They show in their experiments that their technique significantly outperforms LSH.

The set of images that are likely relevant to the query can be quickly established by approximating their feature vectors. In Shyu et al. [2003] principal component analysis is applied to the low-level features of the images and the first few components are used as a filter. Tešić and Manjunath [2003] use vector approximations for a fast nearest neighbor search. First the lower and upper bounds on the distances of each image in the database to the query are computed by looking through the set of all vector approximations. The feature vectors that pass this filter are then visited in increasing order of their lower bounds and the exact distances are computed. An adaptive nearest neighbor approach is proposed that can identify the nearest neighbors for the next iteration based on the feature weights that result from feedback analysis. Heisterkamp and Peng [2005] introduce a kernel vector approximation approach, which allows efficient calculation of upper and lower distance bounds in a kernel induced feature space. Their experiments illustrate the tradeoffs when varying the number of basis vectors and varying the number of bits to represent each basis vector, where more basis vectors and more bits equal tighter distance bounds, while requiring a larger approximation file.

An alternative approach to indexing can be found in Yang et al. [2002, 2004], where some images act as 'peer images' used to index other images. A peer index of an image can be represented as a list of peer images that are semantically related, with a weight attached to each peer image indicating the degree of relevance. There are two levels of indexing, a general peer index that maintains its relevant peer images from the perspective of the whole user community, and a set of personal peer indices that maintain their relevant peer images from the viewpoint of individual users. When the user gives relevance feedback, relevant images are associated with each other and irrelevant ones disassociated at both levels of indexing.

2.3.3 Classification and learning

The core of the retrieval system is the algorithm that learns which images in the database the user is interested in by analyzing the query image and any feedback. Some methods directly assign relevance scores to each image in the database, whereas other methods perform an intermediate classification step and thereafter only focus on the images in the class(es) that are deemed relevant. Most often the latter kind of methods attempt to classify the images using a *one-class* approach, where a model is built for only the relevant class [Chen et al. 2005] or a *two-class* approach, where a model is built that either classifies an image as positive or as negative [Gondra and Heisterkamp 2004]. Other variations exist in order to deal with the restrictions that the one- and two-class approaches pose, for instance *1+x* [Dagli et al. 2006], *x+1* [Hoi and Lyu 2004a], *x+y* [Nakazato and Huang 2002] and *soft label* [Chen and Shahabi 2003]. We illustrate several of these classification variations in Figure 2.4 and show their popularity in Figure 2.5.

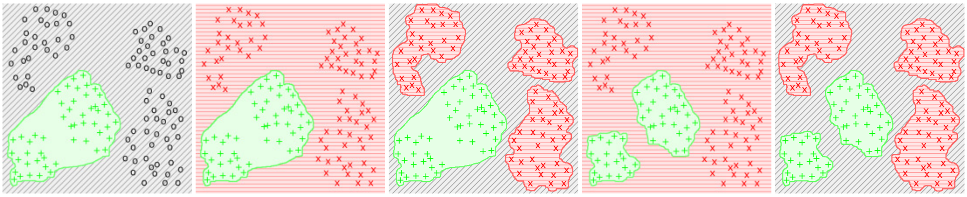


Figure 2.4: Illustration of several classification variations (from left to right): one-class, two-class, 1+x, x+1, x+y. In the figures the green plusses refer to positive examples, the red crosses to negative examples and the black circles to neutral examples. The light green area is then classified as positive, the horizontally striped light red area as negative and the diagonally striped gray area as neutral.

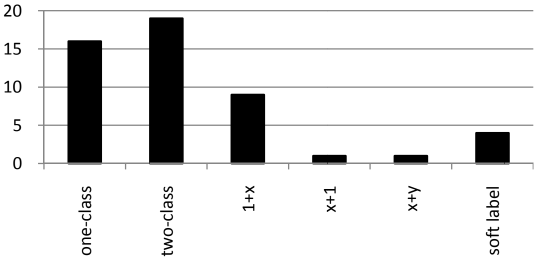


Figure 2.5: Popularity of classification variations in total number of occurrences. Note that multiple variations may have been used in the same paper. Because it is often unclear which classification variation exactly is used, we only counted the variation when it was explicitly mentioned.

2.3.3.1 Query points

A simple approach to finding more relevant images from the user's feedback is based on the assumption that all relevant images are clustered together in low-level feature space. By using a query point movement technique, often inspired by the method introduced by Rocchio [1971], the idea is that the query point is attracted to positive feedback examples and repulsed by negative ones. Ideally after several iterations of feedback the point will be repositioned in the optimal place in feature space where it is surrounded by relevant images. Even though the technique is still being used, its performance seems to have nearly reached its limits. More sophisticated low-level approaches, such as manifold-based learning, or higher level approaches, such as region- and concept-based learning, appear to be more promising. This notwithstanding, current research in this area is ongoing and interesting novelties have been proposed to advance the state of the art. Bayesian decision theory is used by Giacinto and Roli [2002] to compute a new query point that is based on a local estimation of the decision boundary between relevant and irrelevant regions in the neighborhood of the query. The new point is determined by finding the optimal location such that its neighborhood is located as much as possible in the relevant area of feature space. Liu et al. [2006a] propose four target search methods, with the best one following a divide-and-conquer strategy employing Voronoi diagrams to shrink the search space towards the target images, avoiding local maximum traps. The experimental results indicated that quick convergence is achieved even when the initial selected query points are not optimal.

Rather than using only a single query point, retrieval results can be improved by using multiple query points and combining the results. A good discussion of various single and multiple query point movement techniques can be found in Ortega-Binderberger and Mehrotra [2004].

2.3.3.2 Feature selection and weighting

A logical way to discover the hidden information from the user's feedback is to look at the low-level features and let the search mainly focus on those properties that feedback images have in common, for instance by emphasizing the discriminative features. In a sense, feature weighting algorithms can be considered to perform implicit feature selection, because the non-discriminative features generally receive near-zero weights and thus become insignificant after only a few iterations of feedback. Nonetheless, some methods only perform feature selection to find the optimal subset of features [Su et al. 2005] and don't apply feature weighting at all.

An interesting approach is proposed by Doulamis [2007], where each image is represented by a fixed number of bits to keep the retrieval complexity constant. User feedback adjusts the number of bits allocated to each feature based on its relevance to the query. The features are organized into different scales, with each scale being encoded using a certain number of bits. Those features that increase in relevance are expanded into more details by representing them with more bits in the next iteration, whereas the features that are not so relevant anymore are represented with less bits. In Grigorova et al. [2007] a very large number of features is initially extracted from each database. During retrieval a varying subset is adaptively composed by applying predefined rules based on the given feedback, with less discriminating features being removed and highly discriminating features being replaced by a more detailed description. In contrast with the approach of Doulamis, the more detailed features are not more finely quantized, but rather contain an enriched source of information that provides a stronger link between the image and the user's perception. Each of the features of the composed subset are reweighted based on their discriminant ratio [Wu and Zhang 2002b], which is a technique also used in Das et al. [2006] and in Das and Ray [2007]. In the experiments that Grigorova et al. performed the feature adaptive technique improved upon the baseline algorithm and showed comparable results to the more advanced biased discriminant analysis algorithm of Zhou and Huang [2001]. A feature extraction method based on information theory is adopted in Wu and Zhang [2004b], where the entropy, or purity, of the feedback set is calculated involving the percentage of selected relevant and irrelevant images and those features that offer the highest balanced information gain given the entropy are selected.

Huiskes [2005, 2006] represents images by aspects, which are properties that they either possess or do not possess and are defined as conditions on feature values. By analyzing the probability of each aspect occurring in the feedback images and comparing it with the probability of the same aspect occurring in the image database, it can be inferred which aspects the user is interested in. Aspects seem particularly useful in situations where the user selects images that partially contain the topic of interest and they provide the ability to ignore the non-relevant parts of the selected images when searching for similar images. In his experiments the aspect-based technique showed substantial improvement in accuracy over the compared systems, which included the biased support vector machine of Hoi et al. [2004b]. Inspired by the aspect principle, an automatic feature weighting technique is proposed in Thomee et al. [2009a, 2009b] that takes prior feature density into account by giving higher weight to feature value regions where images cluster unexpectedly. This is desirable given that for features to which the user is indifferent, clustering will naturally occur at the feature regions of high prior density and thus the influence of those features is suppressed. A clustering-based approach is also proposed in Chen et al. [2009] for feature selection and weighting. User feedback on images causes constraints to be added that indicate whether or not the images should be considered together, or may not be considered together, for one or more particular features. Both the images and the features are then grouped into separate clusters, from which the optimal features and weights are determined. Goh et al. [2002] follow the idea that a distance function for measuring a pair of images should not be formulated before the images are compared, but rather after they are compared. A dynamic partial distance function is introduced that activates different features for different image pairs. The activated features are called respects, which are those features with minimum differences between them. If a sufficient number of respects are present, then the paired images are perceived as similar. The weights in the distance function are updated by analyzing the feedback. In Nakajima et al. [2003] a feedback technique is proposed that uses the local neighborhood around selected relevant images as additional input. The adjusted query is created by using the differences between the user-selected images and their neighbors to amplify the features that have big differences.

To combat the small sample problem, Wang and Chan [2003b] propose a dynamic sub-vector feature weighting technique. The feature vector is subdivided into multiple sub-vectors, with their dimensions varying according to the number of feedback examples. This subdivision is done through a hierarchical clustering scheme, clustering the feature components into sub-vectors. Each sub-vector is then used as a query to find an improved set of image results and the results of all these queries are fused together by combining the relevance scores assigned to the database images. Stejić et al. [2004] use evolutionary algorithms to tackle the small sample problem, evaluating performance with various sets holding limited number of feedback images. The proposed algorithm randomly generates an initial solution and thereafter iteratively generates new solutions. A new solution replaces the old solution if it is better, as evaluated by the objective function that is based on the ratio of within-class (i.e. positive images) and between-class (i.e. positive and negative images) scatter.

The feature space can be transformed to discover hidden properties amongst relevant images. One of the ways to accomplish this is the well-known Karhunen-Loève Transform, otherwise known as principal component analysis (PCA). This technique is used in Franco et al. [2004], where the positive region is represented by a subspace of feature space, with the assumption that all positive examples belong to this single relevant region. The negative examples are represented in multiple subspaces. The images are then ranked based on their relative distances to the relevant region and the nearest non-relevant region. In their experiments they obtained substantially better results than the algorithms used in the MindReader [Ishikawa et al. 1998] and MARS [Rui et al. 1998] systems. PCA is also used in Tao and Tang [2004a] to obtain the principal subspace and its orthogonal complement. All positive, negative and database images are then projected into the orthogonal complement subspace, with all positive examples ending up at a single location. In this subspace any classifier can then be used.

Another set of techniques for feature selection and weighting are based on linear discriminant analysis (LDA), which analyzes the user feedback to find the most discriminant feature subspace. This subspace is formed by the most discriminant projection vectors so that projected images will simultaneously form the minimum within-class scatter and maximum between-class scatter in the subspace. The small sample problem has a large effect on LDA, since the scatter matrices in practice usually are singular. To deal with the small sample problem the regularization method is often applied [Zhou and Huang 2001], although the computational complexity is quite high due to the high dimensionality of the scatter matrices. Another approach is to reduce the dimensionality first [Belhumeur et al. 1997] to avoid the singularity issue. However then the risk is losing one or more of the most discriminating dimensions from the point of view of LDA in the process. A third approach is to project the examples onto the null space of the within-scatter matrix, where the within-class scatter is zero, after which the optimal discriminant vectors that can maximize the between-class scatter are determined [Huang et al. 2002]. One of the drawbacks of linear discriminant analysis is that negative feedback is treated as belonging to a single class, and therefore research currently focuses on multi-class or biased extensions to improve retrieval performance.

A multi-class approach is discussed in Yoshizawa and Schweitzer [2004], which has as advantage over the two-class approach that less constraints are placed on the projected space, since each negative image is seen as a separate class. Yet, since the negative images are considered to be separate classes, whether or not some of them should actually belong to the same class, the projected space won't be optimal, because the margin between each class is maximized. Therefore, by analyzing the feedback a set of groups can be made that appear to belong together. With new feedback, these groups can grow, shrink, merge and break apart, after which the subspace can be recalculated to obtain improved retrieval results.

Just like its biased support vector machine counterpart, biased discriminant analysis has been developed to give more emphasis to the positive examples and requires the negative examples to stay away from the center of the positive cluster. In addition, the kernel trick can be applied to obtain even better results. Tao et al. [2006a] take things even further by combining the best ideas, resulting in incremental direct kernel biased discriminant analysis (IDKBDA). This approach is a hybrid of LDA, direct LDA, biased LDA and kernel LDA, and is enhanced by an incremental technique to speed up the analysis. In their experiments the IDKBDA method outperformed the algorithms of Zhou and Huang [2001] and Zhang et al. [2001]. In earlier work, Tao and Tang [2004b] argue that approaches that are based on biased discriminant analysis are not able to capture the positive class well, due to their assumption that all positive samples form a single Gaussian distribution. This assumption would be too restrictive for use in content-based image retrieval, since positive images exhibit a lot of variation in image content. Therefore a nonparametric discriminant analysis approach is introduced that relaxes the single Gaussian assumption and additionally requires no parameter tuning. To deal with the small sample problem, a full-space solution is proposed that improves on the null-space method and preserves all discriminant information.

A different kind of approach, discriminant component analysis (DCA), is investigated in Hoi et al. [2006b], which is based on relevance component analysis [Bar-Hillel et al. 2005]. The original approach learns a distance metric by identifying and down-scaling global unwanted variability within the data, and consequently adjusts the feature space so that relevant features are assigned with large weights. Their method now incorporates negative constraints rather than only positive constraints to enhance class discrimination, so that an optimal distance metric is learnt by both maximizing the total variance of data between the negative clusters and minimizing the total variance of data among the positive clusters. In addition, a kernel version is also proposed. In their experiments they demonstrate that both the DCA and the kernel DCA techniques improve over the original RCA technique.

2.3.3.3 Manifold learning

It is well known that search performance generally drastically degrades as the dimensionality increases of the feature space. A myriad of techniques has been proposed to reduce the dimensionality in order to alleviate this problem and make retrieval manageable. The most well-known of these techniques are principal component analysis and linear discriminant analysis, which we covered in the previous section. However, because they are designed for discovering only the global structure of the space, the local structure formed by the query and feedback images is ignored.

Recently research has started to focus on learning this local manifold and the most promising and popular approaches are based on linear extensions of graph embedding. The goal is to create a subspace where the relevant images are projected close together while the irrelevant images are projected far away. This is achieved by embedding the query image and feedback images as data points in a k-nearest neighbor graph, using a weight matrix that indicates the weights on each of the edges. The optimal mapping is found based on this weight matrix, such that neighboring points in the graph are mapped together by minimizing a cost function. Each database image is then also mapped to the manifold. The retrieval results are the nearest neighbors of the query image, and after every round of feedback the manifold is learnt again. Normally not all images in the database are used to construct the nearest neighbor graph. In order to reduce the computational complexity only the top ranked few hundred images from the previous retrieval iteration are used together with all labeled examples.

Augmented relation embedding (ARE) is proposed in Lin et al. [2005] where, in addition to the nearest neighbor graph, two relational graphs are used that encode pair-wise relations in the positive and negative feedback. The weight matrix is constructed in such a way that it is able to cope with the possibility of unbalanced feedback. A closely related method called maximum margin projection (MPP) is proposed in He et al. [2008], which splits the nearest neighbor graph into a within-class graph and a between-class graph. In contrast with ARE, instead of treating labeled and unlabeled images in the same way, two different objective functions are formulated to give more weight to the labeled images. Another similar approach is proposed by Yu and Tian [2006] and is called semantic subspace projection (SPP). This technique uses semantic similarity and geometric similarity as joint constraints to define local neighborhood, instead of letting semantic similarity override geometric information. He [2004b] proposes an incremental version of locality preserving projections (LPP) [He and Niyogi 2003], which he calls I-LPP. Cai et al. [2007b] use the first few steps of LPP in their locality preserving regularized regression (LPRR) algorithm. The set of images and their positive, negative and unlabeled neighbors are gathered into a graph, capturing the underlying local geometrical structure in the data. Cai et al. [2007a] contend that all the aforementioned subspace learning algorithms are linear extensions of the graph embedding approach with different choices of the affinity graph and the constraint graph. To overcome the high computational requirements of these techniques, a unified graph embedding framework is created that performs spectral regression (SR), i.e. regression after the spectral analysis of the nearest neighbor graph. Rather than mapping all relevant points to a subspace where they are located closely together, Liu et al. [2008] use a technique called relevance aggregation projections (RAP) to find the manifold where the query and all relevant examples can be aggregated into a single point, while separating it from all irrelevant examples by a large margin. To overcome several of the problems that current algorithms suffer from a new manifold-learning algorithm, biased discriminant Euclidean embedding, is proposed by Bian and Tao [2010]. Specifically, the method models the intra-class geometry and inter-class discrimination and handles the small sample problem well, in part by involving unlabeled examples in the algorithm.

We can make several interesting observations from analyzing the experiments that all these authors have performed. First of all, it appears that ARE sometimes outperforms LPP, while at other times is the other way around. In the experiments performed by He et al. [2008] they demonstrate that their MPP technique improves upon ARE. The LPRR method of Cai et al. [2007b] is also shown to outperform ARE, and their SR method [Cai et al. 2007a] outperforms LPP as well. He [2004b] shows in his experiments that retrieval performance of his I-LPP subspace is substantially better than in the subspace that is obtained using the original LPP. However, Yu and Tian show that their SPP method improves upon I-LPP. Finally, Liu et al. show that their RAP method outperforms SR, ARE and SPP. Because all experiments have been performed under different conditions it is not possible to draw any real conclusions about which method is the best of them all, though it does appear that both ARE and LPP are being outperformed by their more recent counterparts.

A different approach to learn a manifold is proposed by He et al. [2004c]. First a semantic matrix is inferred from user interaction logs, where the entries are the distances between pairs of images as seen by the users, which should reflect the distances between images in semantic space. Then Laplacian eigenmaps and a radial-basis function network are used to discover the mapping between feature space and the manifold.

2.3.3.4 Probabilistic classifiers

Mixture models are designed to overcome the limitations of using only a single density function to model the relevant class. Using a Gaussian distribution can be very appropriate when the relevant images are all located in the same neighborhood, but when positive examples are further away from the query and are interspersed by negative examples, the relevant class distribution can no longer be modeled by the Gaussian. Mixture models are a combination of multiple probabilistic distributions, where the number of distributions (components) it is comprised of is ideally identical to the number of classes present in the data. The number of classes can be selected from a data point of view, e.g. an estimate of the number of concepts in the database, or from a user's point of view, e.g. one relevant class and the rest negative. In Amin et al. [2007] images are decomposed using wavelets and the wavelet coefficients are modeled by a two-component Laplacian mixture model, using the expectation-maximization algorithm to estimate its parameters. Each database image is modeled by a Gaussian mixture model in Marakakis et al. [2008] and user feedback generates a new positive model by combining the query model with those of the relevant images, where the influence of each relevant model is determined by the relevance degree assigned by the user. In [Tao and Hung 2002] the relevant class is modeled by a mixture of Gaussian distributions determined by the positive samples, while the non-relevant class is assumed to be an average of Gaussian distributions centered at negative feedback samples. Qian et al. [2002] also model the distribution of relevant examples by a Gaussian mixture model, but now using the positive and negative examples for estimation of the model's parameters. Both Amin et al. and Qian et al. compare the performance of their approaches to the approach used by the MARS system [Rui et al. 1998], with the mixture model of Amin et al. showing a large improvement in accuracy, whereas the mixture model of Qian et al. only shows a marginal improvement.

Two Bayesian classifiers are used in Hoiem et al. [2004] to classify sub-images, while performing a windowed search over location and scale to find a user-selected object in query images. The probabilistic model and unconditional density of the positive class must be learned or trained, and to help form this model synthetic images are created from each user-provided training example by translating and scaling it. The second classifier is used to sift out likely negative images that pass the first classifier. In Zhang and Zhang [2004b] the Bayesian theory is used to define the relevancy confidence of a database image in relation to the query image as its posterior probability of being relevant, and similarly for the irrelevancy confidence of being irrelevant. To obtain sufficient negative samples to describe each negative semantic class, with the assumption that each negative feedback example is a separate negative class, hypothetical negative examples are generated based on the class distributions. These hypothetical examples are then used in the estimation of the probability density functions. The overall probability density function for the negative examples is the agglomeration of all individual probability density functions. In Wu et al. [2002a] the probability distribution of all database images is used in combination with the probability distribution of the relevant examples to classify images, where the densities are estimated using a nearest neighbor approach.

A biased minimax probability machine is used in Peng and King [2006a, 2006b], which translates a classification problem into an optimization problem and attempts to determine the hyperplane which can separate two classes of data with maximal probability. The biased machine is an improvement over the standard machine, because it deals better with the imbalanced positive/negative feedback and favors the classification of the positive class over the negative class.

2.3.3.5 Support vector machines

Even though the principles of a support vector machine (SVM) have already been applied in various machine learning techniques since the 1960s and have been used in image retrieval since the early 1990s, the support vector machine nowadays still remains a popular choice for image classification. SVMs aim to find the hyperplane that optimally separates the relevant class from the irrelevant class and is adjusted every time new feedback is received from the user. In its default form the SVM is a linear classifier, but it can be converted into a non-linear classifier by applying the kernel trick to transform the feature space to a higher dimensional space. A separating hyperplane that is found in the transformed space would in effect be a non-linear partitioning of the original space. Proponents of SVMs lauder its good generalization capability, even when only a small set of labeled examples is available. In addition, the solution is always global optimal and absent from local minima. However, opponents quickly point out a number of weaknesses of SVMs. Despite the good generalization with a minimum number of samples, its performance is still quite low. This is partly due to the asymmetric training set that is biased towards the negative examples. Other drawbacks are that overfitting quickly occurs and that tuning the SVM for good performance, i.e. selecting the optimal kernel and its parameters, is a very opaque process and generally is only achieved through trial and error.

Even though at the moment many methods still incorporate a standard kernel-based SVM, the current trend is the development of techniques that aim to overcome their inherent limitations. One approach that targets the imbalanced training set is the biased support vector machine by Hoi et al. [2004b], which is also used by Chan and King [2004]. This SVM uses a pair of spherical hyperplanes in which the inner one captures most of the positive instances while the outer one pushes out the negative instances. A higher weight is allocated to the positive support vectors than to the negative ones. Hoi et al. demonstrate in their experiments that the biased SVM substantially outperforms both a one-class SVM and a soft margin SVM, although the one-class SVM manages to obtain a higher accuracy in the first few iterations.

A different approach to improve performance is the use of soft labels with SVMs. Note that a soft label SVM should not be confused with a soft margin SVM, since the latter type relaxes the condition of finding the optimal hyperplane for separating (noisy) classes by introducing a slack variable that allows controlling the trade-off between maximizing the margin (i.e. potentially ignoring many outliers) and minimizing the training error (i.e. potentially overfitting). On the other hand, a soft label SVM does not force an example to be labeled either +1 or -1, but rather allows them to take on any value from that range. Hoi et al. [2006a] uses the user session logs to obtain the relevance relationships between the images in the database, which are expressed in degrees of confidence. The soft label SVM is proposed, because a normal SVM cannot deal with examples having different confidence degrees. In comparison with the SVM of Tong and Chang [2001], Hoi et al. [2006a] obtained improved classification accuracy in their experiments. In addition, noisy log data had a smaller impact on the performance of their soft label SVM. Rao et al. [2006] suggest a fuzzy support vector machine to handle soft labels. The fuzzy class membership values are used in the SVMs objective function to reduce the effect of less important examples, so that the examples with higher confidence have a larger effect on the decision boundary.

Rahman et al. [2005] use a multi-class SVM technique, where for each pair of classes an SVM is constructed. Given positive and negative feedback, both the dominant positive and negative classes are determined. The database images that are close to the positive class are rewarded, while those close to the negative class are punished. When classifying an unlabeled image the winning class is determined by letting each SVM vote and selecting the one that has the largest number of accumulated votes. Ji et al. [2008] categorize the retrieved images using a pairwise-coupling support vector machine, which is an ensemble of SVMs with each acting on a pair of categories. To retrieve the images the asymmetric bagging support vector machine of Tao et al. [2006b] is used. Similar to the biased support vector machine, the asymmetric bagging SVM uses bagging and bootstrapping to train several classifiers on a balanced number of positive and negative examples. In addition, a random subspace extension is proposed to tackle the small sample and overfitting problems by creating multiple classifiers that each only use a randomly selected small subset of features. This is done to reduce the discrepancy between the number of available examples and the dimensionality of the feature space. Together the asymmetric bagging and random subspace classifiers form a single strong SVM classifier. In comparison with the SVM of Zhang et al. [2001] they not only obtained higher retrieval performance, but also showed that the computational complexity was reduced by a factor of five. An alternative use for SVM ensembles is proposed by Zhang and Ye [2009c, 2009d], where an ensemble is used to filter out noisy feedback. One of the positive feedback images is taken as a prototype and is used to construct a feature dissimilarity space. The ensemble of SVMs is then trained in this space, each using all the positive examples and random groups of negative examples. Once trained, the ensemble reclassifies each positive feedback image and those that it labels as negative are considered to be noisy and are removed.

A proximal support vector machine-based method is proposed by Choi and Noh [2004]. This SVM seeks to find two hyperplanes, called proximal planes, that best represent the positive and negative training samples while maximizing the distance between the planes. In contrast with regular SVMs, where a hyperplane is used for separating the classes, the positive and negative proximal planes lie in the region where respectively most of the positive and negative samples reside. In essence, the positive plane captures what users want to retrieve and keeps it away from what users don't want to retrieve. The distance from the positive proximal plane is then also used as the similarity measure. For increased retrieval speed, a scheme is proposed that uses so-called expanded sets, which involves a collection of images that are related to the initial set of retrieved images. After feedback has been given, only the images in the expanded sets are considered as candidates the user may be looking for. Wang et al. [2005b] also propose a way to reduce the amount of computation necessary between rounds of feedback. By exploiting the knowledge that the confidence in relevance or irrelevance of a sample will at the most change slightly, only the top few most positive samples need to be validated during each round. In their experiments they demonstrated that with a regular SVM the computational time needed for calculating the retrieval results increased per iteration, however with their approach the actual time needed decreased.

2.3.3.6 Artificial neural networks

An artificial neural network attempts to mimic the functioning of biological neural networks, e.g. the way the brain processes information. They are very well suited for image retrieval, because they are able to find patterns in data by learning from example. However they do share some of the drawbacks that also affect SVMs, since they are also prone to overfitting and their functioning is like a black box. Therefore it is hard to properly find the optimal network configuration and initial parameters. In general, relevance feedback adjusts some of the parameters, such as neuron biases and inter-neuron connection weights.

Several different types of neural networks exist and in current research radial basis function network and self-organizing maps are becoming very popular. Yet, the most well-known type, the traditional feed-forward neural network, is still applied to image retrieval. This can be found in the works of Huang et al. [2003a, 2003b], where a three layer feed-forward neural network is used to implicitly perform feature weighting to discover the image region that has the user’s interest.

A probabilistic neural network is used in Ko and Byun [2002] and in Ko et al. [2004] for multi-class learning involving four levels of nodes: input, pattern, summation and output. The user can give four levels of feedback, where each level is represented by a pattern node in the network. Instead of the commonly used sigmoid activation functions, the summation nodes are used to together give the desired discrimination functions between relevant and irrelevant images, yielding a relevance value that can be used for ranking the images.

The radial basis function (RBF) network uses radial basis functions as activation functions, which have the advantage over sigmoids that generally only one layer of hidden radial units is sufficient to model any function. In many works the RBF neural network is used to learn the user's preference by adjusting the centers and widths of the RBF units given the feedback [Muneesawang and Guan 2003, 2004; Jarrah and Guan 2008]. For instance, in Muneesawang and Guan [2003] the positive examples are used to estimate the RBF centers and widths, whereas the negative examples cause the RBF centers to be shifted. The input layer of the network used in Wu et al. [2006] receives image regions and constructs the hidden layer from the feedback images. The output node is then assigned a value which is the weighted combination of all nodes. RBF unit center selection is done by a subtractive clustering algorithm and the estimated centers can belong to either the relevant class or the irrelevant class. Using a probabilistic region weight learning method the relevance is determined of the regions in each cluster center. Qian et al. [2003] use a constructive learning algorithm neural network and overcome some of the network's limitations by using Gaussian-based radial basis functions as the basic neurons in the network instead of the spherical neighborhoods that cover all positive examples. The class of each image can be determined by its membership to each basis function and the learning algorithm automatically determines the number of basis functions to use and their centers and widths by optimizing the covering of the training positive and negative examples by the network.

In contrast with the other kinds of neural networks, the self-organizing map (SOM) network does not need supervision during training. It projects the high-dimensional feature vectors down to only a few dimensions, typically two, through clustering and only two layers of neurons are needed. The map preserves the topology of the data through an iterative training procedure, so that neighboring units in the map contain similar feature vectors. This can be seen in Chan and King [2004], where a self-organizing map is used to partition the feature space and to classify each image into different groups. In Koskela et al. [2002] the map units of several parallel tree structured self-organizing maps are connected with the database images, and each image is connected to the best-matching map unit. Among the images that have a common best-matching map unit, the best-matching image is used as the visual label for that unit. Feedback causes the relevance information to spread to the neighboring units, based on the assumption that similar images are located near each other on the SOM surfaces. The spreading of the relevance values happens by convolving the SOM surfaces with window or kernel functions, see Figure 2.6 for an example. Images with the highest values are then returned to the user.

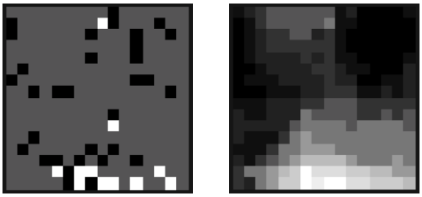


Figure 2.6: The positive (white) and negative (black) map units in a self-organizing map (left) are convolved with a low-pass filter mask, leading to the relevance values being spread across the map surface (right).

2.3.3.7 Kernels

Many approaches use kernels to convert the feature space to a higher- or lower-dimensional space where ideally the images of interest can be linearly separated from all other images. Kernels are commonly used by support vector machines. In Figure 2.7 the popularity of common kernel variations is shown, and as can be seen the Gaussian is the kernel of choice in most research. The kernel that is used is generally fixed, i.e. the type of kernel and its parameters are determined beforehand. However, in recent literature a few methods are proposed where this is not the case. In Doloc-Mihu and Raghavan [2006] the feedback analysis is used to select from a collection of kernels the most suitable one at that point in time for use during the next iteration. This is done by computing for each kernel the score distribution of the positive images and of the negative images, based on the relevance information given by the user. A kernel partial alignment measure is proposed by Zhou et al. [2004] that assists in selecting the optimal kernel and its parameters for performing kernel biased discriminant analysis (KBDA). The aim is to determine the 'alignment' between the kernel and the ideal target matrix on a set of samples, with an emphasis on the biased treatment towards the positive class. The alignment score is used as a goodness measure of the kernel, where a higher score means that there is a better discriminated between classes. Also, multiple kernels can be combined and aligned to give better performance. In Wang et al. [2005a] a suitable kernel is designed after each round of feedback that uses the labeled examples as training data in order to maximize the ratio of the scatter of negative images over that of positive ones. Xie et al. [2004, 2006] aim at learning the user's preference empirically, through probabilistic information that is contained in the user's feedback. This information is then used to derive a kernel that is customized for the specific user and task. In their experiments they show that the retrieval performance substantially increased when using an SVM with their custom kernel as opposed to using the SVM with generic kernels.

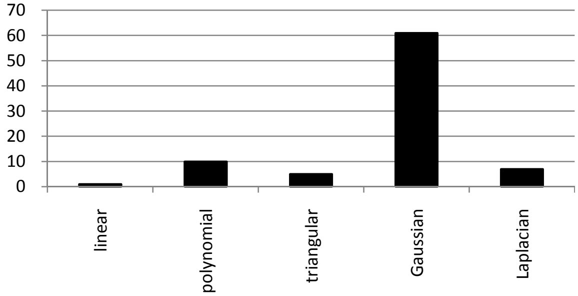


Figure 2.7: Popularity of kernel variations in total number of occurrences. Note that multiple kernels may have been used in the same paper.

2.3.3.8 Active learning

In contrast with passive learning, where the learner randomly selects a few unlabeled examples for the user to give feedback on, active learning is a strategy the learner can apply to speed up learning the concept that the user has in mind. By actively choosing an appropriate set of images, commonly known as the *most informative* images, the user will only need to label a relatively small number of images for the learner to understand the query concept. In comparison, if the user could only give feedback on the best ranking images that are returned after each round of feedback, known as the *most relevant* images, the search converges to a local optimum, because the classifier obtains very little new information. Thus to maximize the information gain for the learner, the best strategy is to select the optimal set of unlabeled examples that minimize the expected future classification error [Bao et al. 2009; Zhang et al. 2009a].

A key issue in active learning is how to measure the information associated with an unlabeled example. One strategy is to use an entropy-based approach to associate entropy values with unlabeled examples, where higher entropy indicates higher information value. In both Jing et al. [2004] and Peng and King [2006b] the probabilities of unlabeled examples belonging to the relevant class are first estimated, after which they are used to derive the entropy values. The min-max framework used in Hoi et al. [2009] assigns a probabilistic value to unlabeled examples that indicates the likelihood of being selected for labeling. Two algorithms are proposed, of which the first solves an optimization problem for all unlabeled examples simultaneously and selects the few examples with the highest values, whereas the second uses a greedy iterative approach to select the most suitable unlabeled examples, one at a time. In comparison with the active learning techniques of Brinker [2003], Hoi and Lyu [2005] and Dagli et al. [2006], which are discussed below, both proposed methods showed a large improvement in retrieval accuracy in the experiments. In Huiskes [2006] those unlabeled images that contain one or more aspects that seem promising, based on their probabilities in relation to their normal likelihood of occurring, are considered to be candidates for labeling by the user. A different approach is proposed in He et al. [2007] and in He [2010], which uses a loss function that takes both labeled and unlabeled images into account, so that the expected errors on the unlabeled images can be properly evaluated. The most suitable unlabeled images are then selected for labeling.

Another strategy is to maximize diversity in the selected informative images. Huiskes [2006] tries to avoid selecting examples with promising aspects that are already overly present in the list of top-ranked images. Dagli et al. [2006] try to achieve the goal of diversity by applying a generalized version of an angular diversity technique, originally introduced by Brinker [2003]. This technique is also used in Chang and Lai [2004] and Goh et al. [2004]. The angle-diversity algorithm calculates a score for each unlabeled example by looking at the angle between the hyperplane associated with this unlabeled example and the currently used separating hyperplane, and those with the highest scores are selected as most informative images. Ferecatu et al. [2004a, 2004b, 2008] use the kernel mapping value between two candidate images to select unlabeled examples, where a low value corresponds to quasi-orthogonality between the images in feature space. This encourages the selection of unlabeled examples that are far from each other. Liu et al. [2007b] perform clustering of the unlabeled examples that are closest to the separating hyperplane and only a single example per cluster is selected to ensure diversity. A similar strategy is employed by Yang et al. [2009], although in this work more than one image per cluster may be selected. In many papers no diversity analysis is performed and simply the unlabeled examples closest to the hyperplane are selected [Hoi and Lyu 2005; Nguyen and Worring 2006; Hörster et al. 2007].

In He et al. [2004a] a criterion is defined that aims to select unlabeled examples that when labeled by the user will maximally shrink the version space. A small version space will guarantee that the predicted hyperplane lies close to the optimal one constructed when all the database images would have their labels. Because evaluating the criterion for every unlabeled example can be computationally very expensive, only unlabeled examples that are close to already labeled ones are considered for evaluation, since the label given by the user will then have a large influence on the new position of the separating hyperplane. Zhang et al. [2008] let the user label only one image at a time, which is the unlabeled example that the learner is most uncertain about. The newly labeled image influences the degree of certainty of the other still unlabeled images in its neighborhood according to their correlation with the image. The next most uncertain unlabeled example is selected for the next round of retrieval. Both He at al. and Zhang et al. demonstrate in their experiments that their new techniques obtain higher retrieval performance in comparison with the active learning approach of Tong and Chang [2001].

2.3.3.9 Combining learners

Instead of using a single learner to classify an unlabeled image, multiple independent learners can be combined instead to obtain a better classification. These independent learners are often weak learners, i.e. they classify images only marginally better than random guessing, but they can also already be strong learners. The idea is that the error of prediction is reduced when the individual learners are combined, resulting in a single stronger learner.

Hoi and Lyu [2004a] combine the decision functions of an ensemble of support vector machines to obtain the final decision function, which is applied to the rankings determined by the SVMs rather than to the class predictions. Multiple Fuzzy SVMs are combined using a bagging-based approach in Rao et al. [2006], where each uses the same positive feedback examples and a different set of randomly selected negative examples. The examples in both sets are chosen to have a small membership value in order to be more informative to the learners. The results are aggregated using weighted majority voting. Bagging and majority voting are also used by Tao et al. [2006b], where a number of classifiers is created by bagging, so that each is trained on a balanced number of positive and negative examples. In addition, a number of classifiers is created by randomly sampling a subset of features to reduce the discrepancy between training data size and feature space dimensionality. The results of all classifiers are aggregated together by majority voting to obtain a single strong classifier. Tu et al. [2008] use bagging to create an ensemble of multiple one-class classifiers, where each attempts to capture the class of interest using a hypersphere. Every unlabeled image can then be labeled based on the average of its probabilities of belonging to each hypersphere. Rahman et al. [2005] use a multiclass technique to construct an SVM for each pair of classes. The winning class is determined by letting each SVM vote and the class wins that has the largest number of accumulated votes.

Rather than combining the outputs of multiple learners in the same way for each search session, Yin et al. [2005] select the most appropriate learner(s) for a particular query or even for a particular iteration. A relevance feedback agent analyzes feedback from multiple user sessions, and pays specific attention to precision rates that reveal the effect from one retrieval state to another state. The agent's goal is to learn an optimal strategy for when to select which set of learners in order to maximize the retrieval performance. They show in the experiments that their technique shows an improvement of roughly 40% in comparison with several query point movement, feature weighting and probabilistic feedback techniques.

A very large set of features, called highly selective visual features, is used in Tieu and Viola [2004], with each feature only responding to a small percentage of images in the database. An approach based on AdaBoost [Freund and Schapire 1997] approach is proposed, where each weak learner selects a highly selective feature for which the positive examples are most distinct from the negative examples. AdaBoost is also used in Huang et al. [2006], where pairs of features are taken by to enhance the learning accuracy. A Bayesian-based weak classifier is trained on each pair and AdaBoost is used to select the best performing ones.

In Wu and Zhang [2004b] a random forest classifier is used, which is a composite classifier that consists of multiple classification and regression trees and involves bagging with random feature selection. To classify an unlabeled example, the random forest lets its trees vote for the most popular class, where each classifier casts a unit vote. To efficiently train the tree classifiers, a biased adaptive technique is used to reduce the size of the training sets and obtain a good balance between the number of relevant and irrelevant examples used. Their adaptive random forest shows improvements over the AdaBoost [Tieu and Viola 2004] technique and an interactive random forest, which is a technique they developed in earlier work. In other work, Wu and Zhang [2004c] train a random forest for filtering out negative images, so that the number of positive examples available to the system is increased in the early rounds of retrieval.

Multiple learners can also be used in active learning to select the most informative images [Singh and Kothari 2003; Zhou et al. 2006; Cheng and Wang 2006c; Zhang et al. 2009b]. Those images are chosen that the learners disagree or are uncertain about.

2.3.3.10 Graph cuts

The graph cut technique is usually used in image segmentation to separate an image into multiple regions. Recent work has applied this technique to image retrieval instead, by using the user's feedback to separate the database into a relevant and an irrelevant group. The feedback images are used as seeds for the graph cut process, with the relevant images forming the source group and the irrelevant images the sink group, and a weighted graph is constructed that models the topology of the database. The partitioning of the database is then performed using a min-cut/max-flow algorithm. Zhang and Guan [2007b] only use the initial feedback to split the dataset and focuses on the relevant group during following iterations with a different relevance feedback technique, whereas Sabhi et al. [2007] repartition the image collection every iteration using the graph cut approach.

2.3.3.11 Synthetic and pseudo imagery

An interesting development is the usage of synthetic or pseudo imagery during relevance feedback to improve the search results. The retrieval system of Aggarwal et al. [2002] uses positive user feedback on one or more regions in the query image to synthesize images containing these regions in different spatial arrangements, see Figure 2.8 for an example. Subsequent feedback on these synthetic images allows it to narrow down what the user is looking for. When query modifications were used in their experiments they obtained a substantial improvement in retrieval performance over when they were not used, in addition to outperforming the well-known methods of Rui et al. [1998] and Rui and Huang [2000]. Hoiem et al. [2004] synthesizes images by translating and scaling the feedback images. The synthetic images are used to assist in creating a probabilistic model of the positive class. In Jing et al. [2003] regions of the query image and positive examples are assembled into a pseudo image, which is used as the optimal query at the next iteration. Similarly, in Karthik and Jawahar [2006] a pseudo image is created and used as the optimal query after every iteration. It consists of the top-ranking segments with small cumulative distance to the positive images and large distance to the negative images. The retrieval system of Thomee et al. [2007b, 2008a] analyzes user feedback and synthesizes new image examples that are constructed to target one or more particular features that appear to be important to the query. The technique is called *artificial imagination* and is inspired by evolutionary algorithms. User experiments showed that the inclusion of synthetic imagery improved the retrieval performance.

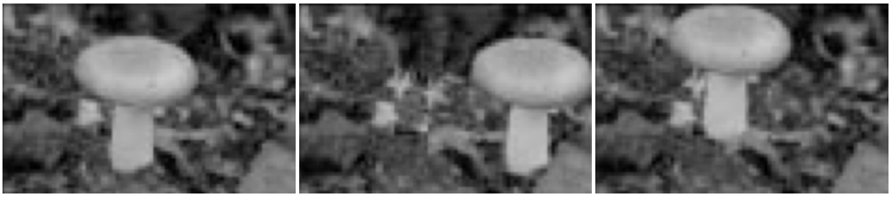


Figure 2.8: Example of synthetic imagery used by Aggarwal et al. [2002], where several images are synthesized containing a region in different spatial arrangements.

2.3.4 Similarity measures, distance and ranking

What truly matters the most in image retrieval is the list of results that is shown to the user, with the most relevant images shown at the top. In general, to obtain this ranking a similarity measure is used that assigns a score to each database image indicating how relevant the system thinks it is to the user's interests. Often the choice of image representation or methodology restricts the kind of similarity measures that can be used. For instance, when Markov random walks are used to construct generative models for the positive and negative class, such as in He et al. [2005], it follows naturally that the similarity measure involves the likelihood functions that both random walks produce, so that the probabilities of being relevant to the query concept can be estimated for the unlabeled examples. However, when images are represented as points in a Riemannian space, which is commonly the case, any similarity measure can be used that is a metric, e.g. Euclidean distance.

The advantages and disadvantages of using a metric to measure perceptual similarity are discussed in Brinke et al. [2004]. The authors argue for incorporating the notion of *betweenness* when ranking database images to allow for a better relative ordering between them.

Si et al. [2006] use the min/max principle to learn an optimal distance metric during a user session. This principle tries to minimize/maximize the distance between the feature vectors of similar/dissimilar images. A regularization mechanism is used to improve the robustness of the metric in case of small and noisy user feedback. Regularization is also used in Tong et al. [2005] to optimize a cost function that combines low-level feature, high-level semantic and current feedback information to produce the relevance ranking.

The relative distance of an image to its nearest relevant and nearest irrelevant neighbors is used as a relevance score in Giacinto and Roli [2003, 2004]. This technique can also be applied in dissimilarity space [Royal et al. 2007] or in a transformed space [Franco et al. 2004]. In later work Giacinto [2007] uses the volume of the minimal hypersphere around the image that encloses the nearest relevant image in the calculation of the degree of relevance.

Multiple similarity measures can also be combined to give relevance scores to the database images. In Rahmani et al. [2005, 2008] an image is seen as a bag of regions and different hypotheses are generated from a randomly selected set of positive bags. The similarity measure is an arithmetic average over the ensemble of the similarity measures given by each hypothesis, using the Hausdorff distance between a hypothesis and a bag. In their experiments they demonstrate that their Accio! system outperforms the SIMPLIcity system of Wang et al. [2001]. In Zhang and Ye [2007a] the p-norm is proposed as an aggregation measure, so similarity functions of different types of features can be dynamically adapted to the user-issued query to optimize retrieval performance. The measure's parameters can be set to not only act like well-known feature aggregation techniques, such as linear or Euclidean combination of the feature distances, but can also be adjusted to emphasize the commonness or difference between features, although the experiments indicated that this did not improve the performance.

In contrast with the similarity measures mentioned above, Wu et al. [2003a, 2003b, 2004a] consider relevance feedback to be an ordinal regression problem, where users don't give an absolute judgment but rather indicate a relative judgment between images. This *m*-rank ordinal regression problem is tackled by decomposing it into *m-1* SVM classifiers and then combining them in a binary tree, where each SVM tries to sift out images belonging to a specific rank and leaves other images to the following SVMs. The final ranking is then obtained by ensuring the images classified to higher rank are presented before the ones with lower rank and using the SVM output values to rank the images within the same rank.

In Figure 2.9 the popularity of common similarity measures is shown. As can be seen the Euclidean (L_2) distance measure is used most frequently, although in a significant number of papers it was only used in the initial iteration and a more advanced similarity measure was applied once feedback was received. As mentioned before, many similarity measures are tailored to the problem to solve and thus quite specialized, and therefore these are not included in the figure.

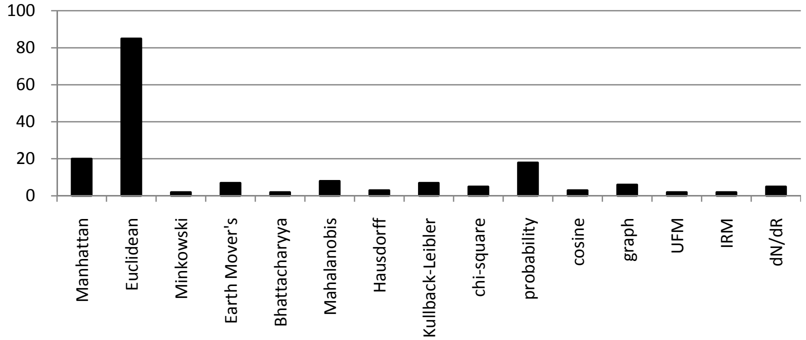


Figure 2.9: Popularity of common similarity measures in total number of occurrences. Note that: i) multiple similarity measures may have been used in the same paper, ii) Minkowski refers to all similarity measures in its family other than Manhattan and Euclidean, iii) probability refers to similarity measures that calculate the likelihood of an image belonging to the target category, iv) graph refers to similarity measures that determine the shortest path between two nodes in a graph, v) UFM means unified feature matching, vi) IRM means integrated region matching and vii) dR/dN refers to similarity measures that use the distance to the nearest relevant image divided by the distance to the nearest irrelevant image as the relevance score of an example.

2.3.5 Long-term learning

In contrast with short-term learning, where the state of the retrieval system is reset after every user session, long-term learning is designed to use the information gathered during previous retrieval sessions to improve the retrieval results in future sessions. Long-term learning is also frequently referred to as collaborative filtering. The most often used approach for long-term learning is to infer relationships between images by analyzing the *feedback log*, which contains all feedback given by users over time. Each entry in the log stands for a single user session, where generally all positive images used in the session are assigned a positive value, the negative images a negative value and all other images the value zero. The information present in the logs is usually aggregated into a so-called *semantic*, *relevance* or *affinity matrix*, which allows the retrieval system to discover to what extent a database image is relevant to the current query [Shyu et al. 2003; Zhou et al. 2003a, 2003b; Qi and Chang 2007]. The modeled relationships can be symmetrical – for instance if image A is relevant when image B is used as a query, then image B will be relevant if image A is used as a query – but this does not have to be the case [Oh et al. 2004]. In Hoi et al. [2006a] the information from the log is used to provide an SVM with more training examples than the user has given by also including the images with the largest relevance scores to the query images.

From the accumulated feedback logs a semantic space is learnt in He et al. [2002], containing the relationships between the images and one or more classes. This matrix is established by applying singular value decomposition to reduce the dimensionality of the feedback log matrix, where ideally the reduced dimensionality captures the total amount of classes in the database. The number of classes must be chosen or estimated, however, and cannot be determined automatically. Similarly, in Shah-hosseini and Knapp [2006] the semantic space is created by applying probabilistic latent semantic analysis to the feedback logs. In their experiments they demonstrated that applying probabilistic semantic analysis gave better results than applying singular value decomposition to learn the semantic space. Another approach to learn the hidden concepts in the database is by clustering the feedback [Chen et al. 2007; Cheng et al. 2008]. In Rege et al. [2007] the images are clustered into one or more image categories, which are placed in a directed graph and may be linked if they appear to be related. New user feedback refines the graph to include new categories or merge existing ones.

A dynamically adjustable distance measure is derived from the log in Yin et al. [2002], which is based on the probabilities that query and database images deliver the same concept. A data mining technique called market basket analysis is used by Müller and Pun [2004]. By analyzing the feedback log the probabilities of associating one image with another can be calculated. These probabilities are then used to predict which features and which weights will return relevant results to the query.

The notion of a *virtual feature* is introduced in Yin et al. [2008], which is long-term relevant information associated on a per image basis. Every time feedback is received, these virtual features are used to find relevant images and adapt over time to changes in user relevance perception. Because an image can contain multiple concepts, the virtual feature records all the concepts that are discovered from user feedback and their significance to the image. In their experiments they showed that the virtual features enabled the system to relatively quickly converge to the ground truth, although many labeled images were required; if a user does not label sufficient images the convergence is significantly slower. In the retrieval system of Barrett et al. [2009] the positive feedback images are considered as a cluster. The semantic clustering is updated by either regarding this positive cluster as a new semantic cluster or merging it with an already existing one.

Chen and Shahabi [2003] use a number of experts to soft-classify the database images. Each user has a profile that contains information about the confidence she has in each of the experts. When a user searches, weights are assigned to the expert opinions based on the confidence values in her profile. The user profile is learnt over time by a genetic algorithm that looks at the feedback that the user gives. In their experiments they compare their expert-based system with MARS [Rui et al. 1998] and achieve a substantially higher retrieval performance. In addition they show that their system is robust to noisy user feedback.

2.3.6 Trends and advances

We can observe from the articles published during the last decade that the perception of image retrieval is slowly shifting from pixel-based to concept-based, especially because it generally leads to an increase in retrieval performance. This new concept-based view has inspired the development of many new high-level descriptors.

Even though the bag-of-words approach has been around for quite a while [Maron and Lozano-Pérez 1998], the technique still remains popular. The same applies to manifold learning [Roweis and Saul 2000; Tenenbaum et al. 2000], which has become a particularly active research area. Several recently proposed algorithms are battling each other, providing a stimulating research environment. Long-term learning and approaches that combine multiple information sources have also demonstrated steady and significant improvements in retrieval performance over the previous years. The inclusion of synthetic imagery during relevance feedback to improve the search results is a completely novel direction, and only time can tell if the technique will gain momentum and attract interest, or be discarded as an attractive but infeasible idea.

2.4 Evaluation and benchmarking

Assessing user satisfaction with the retrieval results is not easy to achieve. Experiments that are well-executed from a statistical point of view require a relatively large number of diverse and independent participants. In our field such studies are rarely performed – recent exceptions being Urban and Jose [2007] and Käster et al. [2006] – although this is understandable due to the rapidly advancing technological nature of our research. Also, in comparison with the medical world where large groups of patients can easily be asked to participate in a new medicine trial, we don't have similar access to such large groups. More often than not our experiments limit themselves to a small group of (computer science) students [Zhang and Zhang 2005a], or do away with real people altogether and use a computer simulation of user behavior [Li and Hsu 2008]. Simulated users are easy to create, allow for the experiments to be performed quickly and give a rough indication of the performance of the retrieval system. However, these simulated users are in general too perfect in their relevance judgments and do not exhibit the inconsistencies (e.g. mistakenly labeling an image as relevant), individuality (e.g. two users have a different perception of the same image) and laziness (e.g. not wanting to label many images) of real users. By involving simulated users, we can very well end up with skewed results. In Figure 2.10 we show how the experiments are evaluated in current research. As can be seen, the majority of experiments is conducted with simulated users, with only a small number of experiments involving real users. Some works provide no evaluation, because they present a novel idea and only show a proof of concept.

A brief look at current ways of evaluating relevance feedback systems of retrieval systems is covered in Marchand-Maillet and Worring [2006] and an in-depth review can be found in Huiskes and Lew [2008a], where guidelines are additionally suggested on how to raise the standard of evaluation. An evaluation benchmarking framework is proposed in Jin et al. [2006], so relevance feedback algorithms can be fairly compared with each other.

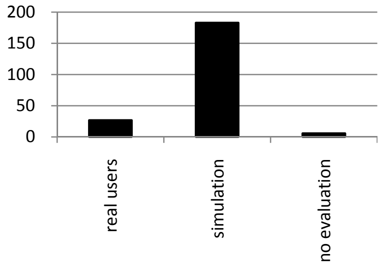


Figure 2.10: Ways of evaluating experiments in total number of occurrences. Note that experiments may have been evaluated using both real users and simulated users in the same paper.

2.4.1 Image collections

There is a large variation in the image databases used by our research community, as is shown in Figure 2.11. It is easy to notice that the majority of research uses the Corel stock image database. The original collection consists of over more than 800 CDs, each containing images of a particular category. Due to its sheer volume the collection is never used in its entirety, which unfortunately has led to the situation that every research group has created their own Corel subset for use in their experiments, e.g. Corel5k [Duygulu et al. 2002]. Because the complexity of image categories varies from one to another, retrieval systems that use different subsets are difficult to compare directly.

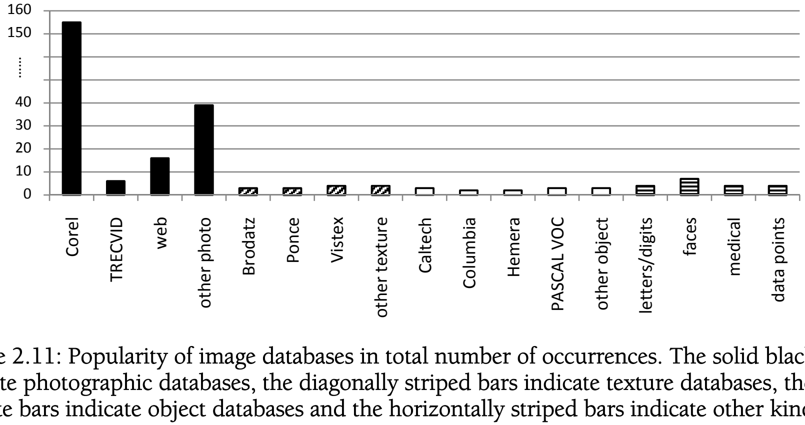


Figure 2.11: Popularity of image databases in total number of occurrences. The solid black bars indicate photographic databases, the diagonally striped bars indicate texture databases, the solid white bars indicate object databases and the horizontally striped bars indicate other kinds of databases. Note that: i) multiple databases may have been used in the same paper, ii) no distinction is made between multiple versions of a database, because it often was not clear which exact version was used (e.g. TRECVid 2003 vs. TRECVid 2005).

The web images used in recent work are often downloaded from Flickr, although Google Image Search and Picsearch are also used as sources. Photographs are not the only images used, because several works use texture (e.g. Brodatz [Brodatz 1966]), object (e.g. Caltech 256 [Griffin et al. 2007]), letter/digit (e.g. MNIST [LeCun et al. 1998]), face (e.g. Yale Face Database B [Georghiades et al. 2001]) and/or medical (e.g. ImageCLEFmed [Müller et al. 2007]) databases. Sometimes experiments are performed on data points that are generated in a high-dimensional space [Hoi et al. 2004b].

The most recent substantial additions to the set of available image collections are the MIR-FLICKR 25,000 [Huiskes and Lew 2008b] and MIR-FLICKR 1,000,000 [Huiskes et al. 2010] sets. Both contain images collected from the Flickr photo sharing website of which all are made available under Creative Commons attribution licenses. These licenses are liberal enough to at least allow the use of the images for benchmarking purposes. This is in contrast to many other collections where the images are copyrighted and officially should not be used. All images include the tags that the original photographer has assigned to them. Additionally, in the 25k image set all images have been manually annotated by several annotators, making this one of the largest image collections of its kind. Since it has only been available to the research community for a very short time it is not yet represented by itself in Figure 2.11. Yet it appears the MIR-FLICKR sets are rapidly increasing in popularity judging from the growing number of citations their papers are receiving.

2.4.2 Performance measures

Recently several new performance measures have been proposed. Huiskes and Lew [2008a] introduce the notion of *generalized efficiency*, which normalizes the performance of a relevance feedback method by using the optimal classifier performance. This measure is particularly useful for benchmarking several methods with respect to a baseline method. Chang and Yeung [2007] assess the retrieval performance based on cumulative neighbor purity curves, which measures the percentage of correctly retrieved images in the *k* nearest neighbors of the query image, averaged over all queries. Figure 2.12 shows the popularity of current methods to evaluate retrieval performance. As can be seen *precision* is the most popular evaluation method, with *recall* second most popular and the combined *precision-recall* as third.

2.4.3 Trends and advances

The calls for standardization are getting louder and during the past years we have witnessed several efforts to fulfill this need, ranging from benchmarking frameworks to standard image databases, e.g. the MIR-FLICKR collections that aim to provide researchers with a large number of images that are well-annotated and free of copyright. Especially now the volume of digital media in the world is rapidly expanding, having access to large image collections for training and testing new algorithms should be beneficial to their quality, especially considering that many current algorithms do not scale well. At present, the Corel image collection remains the de facto standard, despite that there is no standard subset agreed upon for use during experiments.

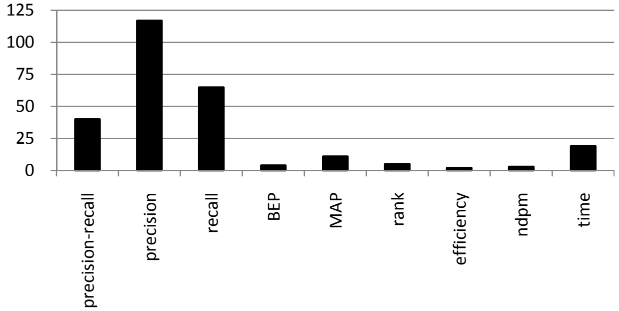


Figure 2.12: Popularity of evaluation methods in total number of occurrences. Note that: i) multiple evaluation methods may have been used in the same paper, ii) precision-recall refers to graphs plotting precision onto recall, iii) precision refers to graphs that plot the precision onto the scope or number of iterations, iv) recall refers to graphs that plot the recall onto the number or iterations, v) BEP means break-even point when precision and recall are identical, vi) MAP means mean average precision, vii) rank refers to evaluations that involve the ranks at which one or more relevant images are found, viii) efficiency refers to the aforementioned generalized efficiency, ix) ndpm means normalized distance performance measure, x) time refers to evaluations that involve the time it took to perform retrieval, xi) satisfaction refers to user-centered evaluations.

2.5 Discussion and conclusions

The rate at which new ideas are born and old ideas are improved is absolutely astonishing. Many of the techniques discussed in this article were not even designed with content-based retrieval in mind. Over the years we have steadily seen the performance of relevance feedback systems get higher, painting a bright outlook for the future. Nonetheless, much research remains to be done. In this section we will discuss the current state of the art and identify open issues, which are problems that our field struggles with and/or that have not yet been adequately addressed. Finally, we will present several grand challenges, which in our view are those issues that must be addressed in order to significantly advance interactive image retrieval, such that retrieval systems will perform better and become more usable, and as a consequence gain traction outside our field.

2.5.1 Issuing a search query, giving feedback and visualizing the results

To the user, a retrieval system is like a black box: a query is inserted, the system thinks about it for a short while, and then retrieval results come out. How this black box exactly functions is not important to the user, as long as (i) the query can be easily formed, (ii) giving feedback feels intuitive, (iii) the images are returned in a timely fashion, and (iv) the images are more or less what the user is looking for. These four issues have been considered the big challenges in the past and remain significant challenges today.

It is important to realize that the user is likely not as well-informed as the researchers that built the retrieval system, and that the users may not understand how to correctly use it, raising for instance the following questions: *what constitutes a 'good' query?* and *what is 'good' feedback?* Such questions are commonly only asked in the context of a proposed retrieval system, but not from a more global point of view. The fundamental issue is that the system should conform to the user, and not the other way around. We have witnessed an immense improvement in usability of systems from the aspect of forming and submitting a query. Compared with not too long ago, when many retrieval systems expected the user to tweak the internal parameters herself [Ko and Byun 2002a], numerous current systems, for instance those described in Section 2.5, hide these internal details, and thus make it easier for the user to use the system. Nonetheless, these improvements in usability have not been properly evaluated in the vast majority of work, an issue that should be focused on in future research.

With the standard way of giving feedback, i.e. asking the user to tick one of several boxes to indicate how relevant she finds each image, it remains unclear to what extent more detailed feedback benefits or hinders retrieval performance. Wu et al. [2004a] show that with four relevance levels better results are obtained than with only two, but this is one of the few works that has actually investigated this topic. The question thus remains what the general effect is on retrieval performance when the user has the ability to give more accurate feedback, and to find out exactly why the performance increases or decreases, and under what conditions. An interesting research direction is to discover how large and to what extent this effect on performance is dependent on the algorithms used by the retrieval system.

The most common choice for displaying the search results is by showing them in a ranked list. Designing alternative ways for visualizing the results have only recently attracted significant attention, which may be explained by great improvements in application development environments over the last few years, which facilitate the design of interactive interfaces. As is the case with issuing a query and giving feedback, the visualization of the search results also requires more research, in particular how the results can be presented so that the arrangement of the returned images feels natural to the user. An interesting direction to probe into is the (automatic) categorization of image results, somewhat similar to the concept ontology presented by Fan et al. [2008], albeit in a relevance feedback setting.

Answers obtained from investigating all these issues can facilitate broader acceptance by users, because the search engines will become more intuitive and usable. In general, more attention should be directed towards user-friendliness when any part of a retrieval system directly interacts with the user. We strongly encourage collaboration between researchers in our field with researchers in the area of human psychology to work jointly on further improving human-computer interaction.

2.5.2 Retrieval algorithms

We can clearly observe how relevance feedback-based learning algorithms have evolved from the early days into their current form. Experimental results that compare the latest learning algorithms with those used in systems from the previous millennium demonstrate that retrieval performance has improved substantially. Of course, the problem of finding the images that the user is looking for is far from being solved, but our field is definitely moving forward. We can separate the current state of learning algorithms into the following three areas.

Classic learning algorithms

These techniques have been around for quite a while and generally perform well, but have reached the point where current advances are very limited in scope. Most techniques that operate in low-level feature space belong in this category, such as query point movement and feature weighting. Also in this category are minimum distance classifiers and Hebbian neural networks, which are hindered by their intrinsic drawbacks, in particular the difficulty they have with a small number of training examples.

Promising learning algorithms

These techniques are relatively young and have demonstrated over the past years that they work well and that they still show much room for improvement. Both active learning and long-term learning fall into this category and incorporating either of them is almost a guarantee to improve retrieval performance. Kernels have received a large increase in attention in the last decade, partly due to the realization that they can be applied in many different areas and not just in support vector machines. It is unclear how much impact kernels can still have in our field, making it worthwhile to look further in this direction. Manifold learning is also a technique that looks very promising, this in contrast with most other methods that operate in low-level feature space and belong in the category above. The field of manifold learning is very lively and many new methods have been proposed during the last few years. The highly competitive spirit of the researchers is apparent in their papers, where they actively compare their manifold learning techniques with each other. It will be interesting to see in the future how much improvement manifolds are able to achieve. Combining knowledge sources is another very active and promising direction, and its potential is strengthened by the positive results obtained by current work on, for instance, fusing textual and visual features together.

Novel learning algorithms

Novel learning algorithms are being regularly developed in the machine learning and the neuroscience fields. A particularly interesting direction comes from spiking networks and BCM theory [Baras and Meir 2007], which arguably is the most accurate model of learning in the visual cortex. Another novel direction is that of synthetic imagery. At the moment, learning algorithms lack the ability to ask the user specific questions on uncertain aspects of the query. Active learning comes close, but is restricted in asking for feedback on images in the collection. The use of synthetic imagery can have a significant positive impact on retrieval performance. One example that many of us may be familiar with is the game "20 Questions", where the goal is to figure out which particular subject, object or concept the other player has in mind. This is achieved by asking questions that must be answered by a simple 'yes' or 'no', for instance do you have a person in mind? and can I eat it?, and narrowing the possibilities down towards the solution. By letting the learning algorithm whittle down its questions about particular image characteristics, it should be able to figure out much sooner what the user is looking for than through conventional approaches. This technique is not restricted by existing images in the database; rather it can synthesize completely new images and ask the user for feedback. At present this technique is still in its infancy, but its outlook is very promising.

Utilizing implicit feedback to aid a learning algorithm in discovering what the user is looking for can be another way to boost retrieval performance. Typically a user does not enjoy giving feedback, thus any bit of useful information that can be freely obtained and can have a positive impact on finding out the user's intentions ought to be analyzed. Furthermore, the user does not enjoy waiting too long for the results to show up. Fortunately, those approaches that give excellent retrieval performance, but are too computationally intensive at present, will eventually become acceptable to the user as technology advances.

2.5.3 Image representation

An area in which much progress has been made is how to represent images for use in relevance feedback. Research no longer solely concentrates on low-level feature spaces, but rather attempts to mimic the way people look at images by approaching retrieval from a more semantic point of view. Region-based and concept-based retrieval have shown very high performance over the last years, and these directions should certainly be further explored in the years to come. Techniques that are based on interest points, for instance SURF [Bay et al. 2008] and SIFT [Lowe 2004], are very popular in non-interactive image retrieval systems because they show excellent retrieval performance, but have not frequently been applied to interactive retrieval. One recent work is that of Rahmani et al. [2008], although in their experiments the interest points-based region segmentation approach did not yet lead to improved results, in comparison with their segmentation approach from earlier work [Rahmani et al. 2005]. Still, it may be very well possible to realize the high performance of interest points in interactive image retrieval, making this a promising research topic.

2.5.4 Experimentation

The majority of experiments are performed with simulated users. While simulations are very useful to get an initial impression on the performance of a new algorithm, they cannot replace actual user experiments since retrieval systems are specifically designed for users. In fact, as we stated before, we can very well end up with skewed results by involving simulated users instead of real users. Even though it is difficult to find a diverse set of users willing to participate in our experiments, we should still strive for the involvement of real users, in particular because developments in our field take place rapidly and we therefore have to perform experiments frequently.

For assessing the performance of a system, we observed that precision- and recall-based performance measures are the most popular choices at the moment. However, Huijsmans and Sebe [2005] discovered that these measures are unable to provide a complete assessment of the system under study and argue that the notion of *generality*, i.e. the fraction of relevant items in the database, should be an important criterion when evaluating and comparing the performance of systems.

Interestingly, when similarity between images is determined, there is almost never any justification to be found why a particular similarity measure is chosen. It would be worthwhile to investigate this further to understand the tradeoffs involved when selecting a particular similarity measure and its effect on retrieval performance, and also how much the resulting performance depends on the descriptors used.

2.5.5 Standardization and reproducibility

The large variety of available image collections makes it difficult to compare methods with each other. The call for standardized image collections is frequently heard, for instance in Müller et al. [2002], where the authors also demonstrate how easy it is to change the apparent performance of a retrieval system without adjusting the system and even when using the same Corel image collection in the experiments. During the past ten years the content-based retrieval community has been increasing the size of the credible test sets, from hundreds to thousands. It has repeatedly been found that methods that work very well on a thousand images frequently perform poorly on ten thousand images. For scalability purposes it is thus important to have access to very large databases. From both a standardization and scalability point of view, the MIR-FLICKR image sets may prove to be a step in the right direction.

Experimental results need to be placed in context with already existing techniques to support any claims of advancing the state of the art. It is certainly possible for a researcher to implement the techniques proposed by other researchers, so they can be compared using the same conditions, but in practice this approach does not work due to (i) the amount of time it takes to properly implement another technique and having to ensure that the implementation is correct, or (ii) because the publicly available source code of the technique is not straightforward to compile and binaries are not available or target a different operating system. These difficulties hinder our community and ways should be found for making the implementations of techniques publicly available in a common format.

2.5.6 Emerging technologies and trends

Recent developments in the industry have led to touch-based technology no longer having a niche status but having gone mainstream. These developments open up new interaction possibilities between search engine and user. With the market share of fast, touch-enabled smartphones rapidly increasing, novel interfaces can be created that deliver a better search experience to such devices, while at the same time reaching a large number of users.

The current mindset of cloud-based computing and greater availability of high-bandwidth internet connections have spurred research on distributed search systems. Considering that networking speeds are constantly getting faster and image collections larger, distributed techniques may be the key to keeping response times acceptable.

Now that the Web 2.0, the social internet, is becoming more and more prevalent, techniques that analyze the content produced by users all over the world show great promise to further the state of the art. The millions of photos that are commented on and tagged on a daily basis can provide invaluable knowledge to better understand the relations between images and their content.

2.5.7 The future of relevance feedback-based image retrieval

One might expect the search giants, such as Google, Microsoft and Yahoo!, to have embraced a relevance feedback-based image search engine, since it could lead to both more satisfied users and increased revenues. The nature of relevance feedback would allow them to guide the user through multiple pages, serving them with advertisements along the way that generate profits. Since the search giants have not yet incorporated relevance feedback, we ought to investigate what the main issues are that hinder widespread adoption.

Even though we do not have a crystal ball that foretells the future of relevance feedback, the past decade has brought us many exciting new developments and advances that are bound to inspire further research. With technological progress occurring rapidly, there is no doubt relevance feedback will benefit and also make rapid progress in the coming decade. In conclusion, the most pressing grand challenges can be summarized as follows:

User interface problem – What is the optimal user interface for queries and results?
Our current systems usually seek to minimize the number of user labeled examples or the search time on the assumption that it will improve the user satisfaction or experience. A fundamentally different perspective is to focus entirely on the user experience. For example, it is possible that the user's overall satisfaction level regarding a search session could be higher if the experience is enjoyable. We could potentially turn the search session into a game or a fundamentally different interface which could have a significant positive impact on the user experience. A longer search time might be preferable if the overall user experience is better.

Small training set problem – How can we achieve good accuracy with the least training examples?
The most commonly cited challenge in the research literature is the small training set problem which means that in general the user does not want to manually label a large number of images. Developing new learning algorithms and/or integrating knowledge databases which can give good accuracy using only a small set of user labeled images is perhaps the most important grand challenge of our field.

Evaluation problem – How should we evaluate and improve our interactive systems?
Evaluation projects in relevance feedback-based image retrieval are in their infancy. Currently, most researchers attempt to use simulated users to test their algorithms, knowing that the simulated behavior may not mirror human user behavior. How should one model the simulated user and which image collections and ground truth should be used? Finally, the evaluation projects should seek not only to determine comparative performance benchmarks but also to give insight into each system's weaknesses and strengths.

3. Artificial imagination

We propose a novel retrieval technique that we call artificial imagination, which gives the search engine the ability to 'imagine' by synthesizing images. Our aim is to determine if such synthetic images can be beneficial to visual search. We present an evolutionary algorithms-inspired method for synthesizing textures.

3.1 Introduction

The definition of "imagination" according to the Merriam-Webster dictionary is:

> **imagination**: the act or power of forming a mental image of something not present to the senses or never before wholly perceived in reality.

This definition of imagination forms the foundation of a paradigm we call *artificial imagination* (*AIm*). Analogous to the concept of artificial intelligence, where the computer is given the ability to be 'intelligent', we intend to give the computer the ability to 'imagine', where the meaning directly ties into our ability to synthesize images of objects or of the world that do not have to conform with reality. Another important analogy with artificial intelligence is the perspective of intelligence lying on a continuum from a simple game playing program to the highest level of intelligence, which arguably would be human intelligence. Similarly, artificial imagination also lies on a continuum where the highest level may be the human imagination, and the lowest level might be a simple random image generator. Imagination in regards to synthesis has been given significant attention in the field of computer graphics where synthesized imagery is the norm, see for example Figure 3.1. It is important to note that such images do not need to have any functional usage, since they could simply be artistic, see for example Figure 3.2.

Figure 3.1: Synthetic imagery in movies: a scene from Wall-E.

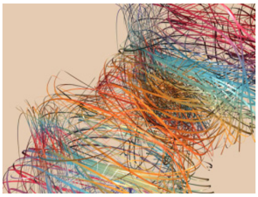


Figure 3.2: Synthetic imagery in art: a piece made by Charles Csuri.

We are not only looking at possibilities for synthesizing images, but we are especially interested in exploring whether such imagined images can be beneficial in the context of content-based image retrieval. We are targeting an approach where we think synthetic imagery can result in a significant improvement of the number of relevant images returned to the user. In Section 3.2 we will discuss this approach and we present our experiments in Section 3.3. Finally, we conclude in Section 3.4.

3.2 Synthetic imagery

When we are learning new visual concepts, we often construct new mental images that are synthesized from our imagination and that serve the purpose of clarifying or helping us understand the primary features that are associated with the visual concept. One example from real life is when a police officer creates a sketch of an unknown person to help identify a thief and she asks the victim certain questions to describe this unknown person more clearly, such as "what hair color did the thief have?" and "did the thief have a big or a small nose?". Artificial imagination can be seen as the digital analogy of our own visual imagination. In the context of relevance feedback-based image search, the police officer would be the search engine, the victim would be the user, and the thief would be the target image the user has in mind.

Our idea is that the search engine can synthesize images that target one or more particular image characteristics that appear to be important to the user, with the intention of clarifying exactly what the user is aiming to find. In this case, artificial imagination can be considered as an advanced type of active learning, since the systems asks the user specifically for feedback on a particular example in order to meet certain informational needs as well as possible. In the artificial imagination paradigm examples are not restricted to be taken from the database itself like in traditional active learning, but are rather synthesized to more directly satisfy these informational requirements. To illustrate, envisage a future situation where the search engine is not sure whether the user is searching for an image containing cows, or for one containing sheep. The ability to synthesize two images where one shows cows and the other shows sheep will give the search engine more clarity of the user's interests once it has received feedback on them. We attempt to lay the foundations for such a retrieval system with the techniques proposed in this chapter.

Potentially there are many techniques towards generating synthetic examples. Image synthesis can be performed on a statistical level [Simoncelli and Portilla 1998], or directly by using transformations such as the Karhunen-Loève transform [Therrien 1989]. An issue to consider is that not all image descriptors are equally suited for image synthesis because image reconstruction is generally ill-defined: a one-on-one mapping from an image descriptor to an image may not exist, see Figure 3.3 for an illustration. This is problematic from the point of view that descriptors that are very appropriate for image retrieval may not be appropriate for image synthesis, and vice versa. As a straightforward and effective solution, we propose to use two different image descriptors in our method, one that will be used for retrieving images and the other for synthesizing images. Each database image will thus be associated with two image descriptors.

If we restrict ourselves to techniques that use low-level image features (e.g. colors, edges), we can then understand artificial imagination as the intelligent synthesis of examples based on feedback images and their locations in feature space. Our aim is to examine their relevance in relation to the other database images and attempt to find locations that are key to clarifying uncertain or emphasize important features. We can then synthesize images based on these locations in feature space and present them to the user for feedback. Our assumption is that asking the user for feedback on these constructed examples will benefit the retrieval results in two ways: (i) any feedback given by the user on these images allows the system to obtain a better understanding of what the user is looking for than it would from feedback on database images alone, and (ii) the amount of iterations necessary to satisfy the user is reduced, since the target image(s) will be found sooner. Once the user gives positive/negative feedback, the retrieval system uses the first feature space (FS_R) for retrieving an improved set of images and the second space (FS_S) for synthesizing images. Once an image has been synthesized, its missing FS_R feature vector can easily be calculated by treating it as if it were a regular database image.

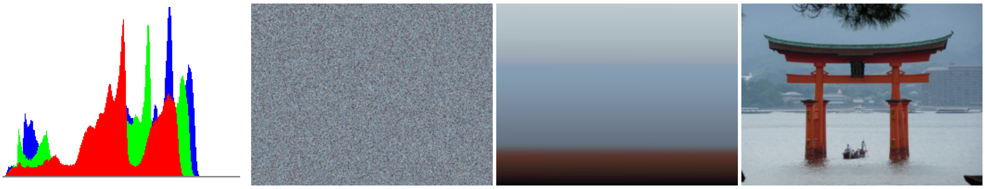


Figure 3.3: Image reconstruction from a point in feature space is generally ill-defined. From a particular color histogram (left) it is possible to synthesize a large number of valid images.

An approach inspired by evolutionary algorithms is very well suited to determine the optimal locations in feature space for synthesis since such algorithms take a population and evolve it towards better solutions. In our situation this leads to evolving a population of images towards a solution ideally containing all images of interest to the user. Our algorithm has four steps: starting population, crossover, mutation and survival, and after the last step the algorithm loops back to the second step. The algorithm is defined as follows:

1. *Starting population.* Let $R_0 \subset \{1, \cdots, n\}$ be a random subset of images from the database of size $N_R \leq N$ and show them to the user. The positive feedback gives the initial population S_1^+.

2. *Crossover.* In this step we sub-sample S_t^+, the positive examples from iteration t, using their feature vectors in feature space FS_S. Consider sets $A_j \in 2^{S_t^+} : |A| \geq 2$ containing at least two positive images from the feedback, with $j = 1, \cdots, N_j$ and $N_j = 2^n - 1 - n$. Each of these sets gives a new query point $c_j = \frac{1}{|A|} \sum_{i \in A_j} x_i$. For step 3 we use the set $C = \{c_1, \cdots, c_{N_j}\}$.

3. *Mutation.* We perturb the points generated in the crossover step by two mechanisms. First we use the negative feedback of iteration t to push away points towards a more favorable area in feature space, and then we introduce random elements. Let $\bar{x}^{-} = \frac{1}{n^{-}}\sum_{i\in S_t^{-}} x_i$ be the mean of the negative feedback. We use this for each query point in C to obtain mutation points $c_j' = (1-\delta)c_j + \delta\bar{x}^{-} + r_j$, with $0 \leq \delta \leq 1$ and r_j small random values. Finally, we synthesize images from the mutation points.
4. *Survival.* The synthesized images, together with the improved set of images R_t resulting from applying a retrieval algorithm in feature space FS_R, are presented to the user. We let the user determine which images are most relevant and survive into the next population S_{t+1}^{+}.

In Figure 3.4 and Figure 3.5 we give an illustration of the usefulness of synthetic images. If a user is looking for crosshatched images, while only an image containing horizontal lines and an image containing vertical lines are shown on screen, the algorithm is able to synthesize a crosshatched image (Figure 3.4). Alternatively if the user is looking for a color adjusted version of an image the algorithm can synthesize an appropriate image (Figure 3.5). We expect the search to be steered more quickly into the correct direction if the synthetic images are used in queries.

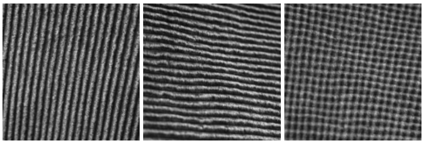


Figure 3.4: Synthesis of a crosshatched image (right) from images containing vertical lines (left) and horizontal lines (middle).

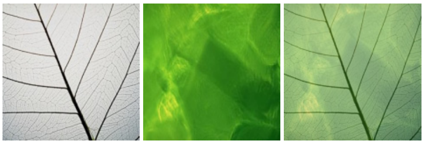


Figure 3.5: Synthesis of a color adjusted image (right) from textures containing the pattern (left) and the adjustment color (middle).

3.3 Experiments

In our experiments we use the Ponce texture collection [Lazebnik et al. 2005] as the test database, because it is standardized, well-known, easily available and considered to be challenging by the texture retrieval community. Our 3000 image test set included the 1000 original textures and 2000 images that were either randomly rotated or scaled from the original versions by up to 15%, resulting in a set of textures that vary in 3D perspective, shape and orientation. For the retrieval feature space we use the MPEG-7 homogeneous texture descriptor [Ro et al. 2001], because it is specifically designed for texture retrieval. For the synthesis feature space we use the aforementioned Karhunen-Loève transform (KLT). We create this feature space by taking l coefficients from a KLT representation of the image collection I, thus $FS_S = \{x_i \in \mathbb{R}^l | i = 1, \cdots, |I|\}$. When the feature space relevance analysis indicates that a particular point in feature space is of significant interest, we note that the point in feature space is a set of coefficients for the eigenvectors in the KLT representation. We can then create the corresponding image by the standard method of linear reconstruction using the coefficients and corresponding eigenvectors, i.e. given a feature vector x containing the coefficients, a new image s can be synthesized through $s = Wx + \mu$, where the columns of W represent the eigenvectors of the KLT and μ is the average image used for denormalization. Thus all texture images are represented by both MPEG-7 features and KLT coefficients.

Our goal is to measure the effectiveness of the synthesized images in the relevance feedback process. We have therefore implemented two algorithms: a standard algorithm ('Standard') and an enhancement of the standard algorithm where synthesized images are introduced ('Synthetic'). For both algorithms we chose the well-known Rocchio [1971] method to operate in the retrieval feature space for obtaining an improved set of images. Rocchio's method takes the current query point in feature space and uses the feedback given by the user to move it towards the positive images and away from the negative images:

$$q_t = \alpha q_{t-1} + \beta \left(\frac{1}{n^+} \sum_{1 \in S_t^+} x_i \right) - \gamma \left(\frac{1}{n^-} \sum_{1 \in S_t^-} x_i \right), \tag{3.1}$$

where $S_t^+, S_t^- \subset \{1, \cdots, |I|\}$ are the index sets of the positive and negative points, respectively, at iteration t; n^+ and n^- the size of the positive and negative index sets, respectively; q_t and q_{t-1} are the new and current query points, respectively, and α, β and γ are suitable constants.

For our experiments, which involved 30 students, we implemented blind user testing to minimize bias, i.e. the students were not aware which version (Standard or Synthetic) of the algorithm they were using. The students were assigned 6 queries each: 3 image queries and 3 text queries. With an image query, the user was given an example image and was asked to find similar images. With a text query, the user was given a texture category and was asked to find images that would fit that particular category, e.g. 'marble' or 'tree bark'. The categories were selected from Getty Images stock photography keywords, the Ponce texture classes, and several were suggested by users we worked with previously, see Table 3.1 for an overview. In total, there were 20 different image queries and 20 text queries that were randomly assigned to the users.

The students were asked to record the number of images they considered to be relevant at iterations 1, 4, 8 and 12. Per iteration 15 images were shown on screen, where in the Standard algorithm all images shown were from the database, while in the Synthetic algorithm some of the images were synthetic (given n positive images $2^n - 1 - n$ synthetic ones, with a maximum of 3) and the remaining ones from the database. At iterations 1, 4, 8 and 12 only the top ranking database images were shown, thus no synthetic ones, in order to enable a fair comparison between the two methods. The most relevant results to stock photography are shown in Figure 3.6 and the overall average combined text and image precision results in Figure 3.7. Here, we consider precision as the number of relevant images found over the top 15 best ranking images.

Table 3.1: Texture categories

Ponce	Getty Images	User-based
brick (fine)	horizontal composition	old stone
brick (coarse)	vertical composition	curvy/organic look
tree (bark)	abstract	wavy
tree (wood)	sparse	diamond pattern on fabric
marble	square/rectangular	tartan pattern
fabric	rustic/rural/country	linear
	man made/synthetic	
	nature/natural	

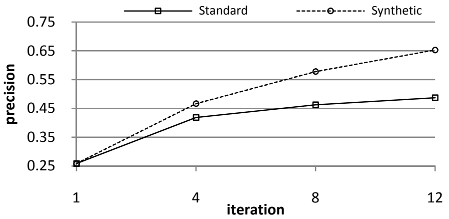


Figure 3.6: Getty Images text query results.

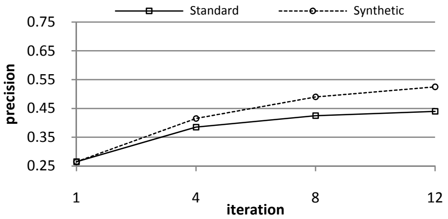


Figure 3.7: Average text/image query results.

As can be seen, the Synthetic method compared with the Standard method shows a constant improvement in the amount of relevant images shown on screen. Our results thus indicate that the inclusion of synthetic imagery leads to an improvement of the amount of relevant images shown to the user. If we look at the separate image query results, the Standard method obtained a precision of 0.40 at iteration 12, whereas the Synthetic method had a precision of 0.46, which is an improvement of 15%. For the text queries, the Standard method had a precision of 0.48 and the Synthetic method achieved 0.59, which is an improvement of 23%. In both cases the Synthetic method outperformed the Standard method.

3.4 Conclusions

Artificial imagination is a promising paradigm that can significantly enhance the level of interaction of the retrieval system with the user. Based on user feedback, our method uses an evolutionary algorithms-inspired technique to synthesize images that are constructed to clear uncertain or emphasize important image features. From additional feedback on these images, the search engine is better able to determine what the user is looking for. The results of our experiments indicate that by giving the retrieval system the power to imagine, it will more quickly understand what the user is looking for.

However, unlike general images, textures do not necessarily have any semantic meaning. Modifications to a texture generally result in a new valid texture, which makes our current approach very appropriate for texture search. Because our method has no knowledge of semantic concepts, e.g. it cannot synthesize an image containing a dog on a beach given an image of a beach and another with a dog, it is not particularly suitable for general image search. The main future challenges thus lie in the design of meaningful methods of image synthesis, for example the generation of collages, where the system can combine image concepts or objects by placing them in a single image. Alternatively, an image can be synthesized where the focus is not on the perceived realism of the image, but rather on the spatial layout of image elements (objects), enabling the search engine to search by image composition instead of by exact visual similarity. The semantic image synthesis technique of Hays and Efros [2007] has been shown to create plausible scenes by seamlessly blending together appropriate image regions found in the image database. We are currently focusing on integrating this technique into an interactive image retrieval system. The search engine allows the user to interactively erase unwanted parts of images and have them replaced by content that the user is interested in. The idea is that these touched up images will lead to better retrieval results, because they more closely match what the user is looking for. We present our early work on this search engine in Appendix B.

4. Visual exploration and search

Experiential retrieval systems aim to provide the user with a natural and intuitive search experience. The goal is to empower the user to navigate large collections based on her own needs and preferences, while simultaneously providing her with an accurate sense of what the database has to offer. In this chapter we integrate a new browsing mechanism called deep exploration with the proven technique of retrieval by relevance feedback. In our approach, relevance feedback focuses the search on relevant regions, while deep exploration facilitates transparent navigation to promising regions of feature space that would normally remain unreachable. Optimal feature weights are determined automatically based on the evidential support for the relevance of each single feature. To achieve efficient refinement of the search space, images are ranked and presented to the user based on their likelihood of being useful for further exploration.

4.1 Introduction

Over the past years we have seen image collections grow tremendously, both in a personal sense (e.g. home photo collections) and in a public sense (e.g. image databases on the internet). As a result, finding images of interest has become more and more like finding a needle in a haystack. As we have seen in Chapter 2, recent advances in retrieval techniques have progressed the state of the art significantly, but they have not yet been able to solve the general problem of image retrieval. Notwithstanding the diversity in the techniques that are currently used, the general consensus is that incorporating relevance feedback leads to improved search results and therefore has been applied in the majority of research from the moment the concept was introduced by Rocchio in 1971.

One of the grand challenges in our field is considered to be the need for experiential exploration systems that allow the user to gain insight into and support exploration of media collections [Lew et al. 2006]. For users, exploration is the predominant mode of interaction, rather than querying, and therefore interfaces that accommodate for this behavior are needed [Jain 2003]. In this chapter we propose such a system, where the user can visually explore the feature space around relevant images and focus the search on only those regions that are relevant. Each of these regions centers on a relevant example image and is bounded by its relevant nearest neighbors. In addition, the user can effortlessly navigate from one area in feature space to another to discover more relevant images using a technique we call *deep exploration*.

The underlying set of features used to describe the database images is generally considered to be one of the most critical aspects for retrieval performance and consequently user satisfaction. Having a large set of features is not a guaranteed way to correctly retrieve all desired images. Rather, increasing the number of features makes it computationally more intensive and storage-wise more expensive for the system, and also causes the retrieval performance to suffer from the so-called curse of dimensionality [Böhm et al. 2001]. At the same time, a small but inappropriate set of features will not produce the results the user is looking for, because they cannot capture the user's intentions well enough. Since in low-level feature space the images of interest can be scattered over multiple areas, feature selection and weighting is a sensible way to transform the feature space so that these images that are perceptually close to each other are also close to each other in the resulting space. Note that feature weighting schemes implicitly perform feature selection, because the weights for one or more features usually go down to zero after only a few iterations. Many retrieval systems analyze the feedback given by the user to figure out which image features are important and also how important they are. In our system the images, contained within the regions that the user has focused on, are used to automatically determine the optimal set of feature weights for an image when it is explored, based on the evidential support for the relevance of each single feature. The images are then ranked using their associated feature weights in two ways, where one ranking reflects the likelihood the image is useful for further exploration of the feature space and the other reflects the likelihood the image is relevant to the user.

In Section 4.2 we will first look at related work, and we discuss the proposed exploration technique in Section 4.3. The feature weighting approach is covered in Section 4.4. Section 4.5 describes the experiments we performed and we conclude in Section 4.6.

4.2 Related work

Despite its potentially great impact on user satisfaction and retrieval performance, the interface is often still a largely ignored component. Most work only focuses on how to present the search results [Datta et al. 2008], whereas work on visualization for assisting the user to search more efficiently is rather limited. In Nguyen and Worring [2008] a similarity-based visualization technique is used to project the image collection onto a 2D manipulation space, where the user can easily select groups of similar images together and refine the selection. Hyperbolic visualization of a concept ontology is used in Fan et al. [2008], allowing the user to obtain an overview of the image collection at a concept level and to interactively navigate the concept ontology by zooming in on different concepts of interest. The notion of visual islands is introduced in Zavesky et al. [2008] to fulfill the principal goal of guided user browsing. This includes a process called island hopping that is used to dynamically reorganize the displayed pages according to the user’s selection, so that the user can explore deeper into a particular dimension that she is interested in.

While the user searches and gives feedback, many retrieval systems attempt to figure out which image features are important to the user and also how important they are. Over the years, feature weighting and selection techniques have therefore received much attention. The well-known AdaBoost algorithm is used in Tieu and Viola [2004] to reduce a feature set containing thousands of highly selective features to a subset consisting of only those features that are highly discriminating for the given query. In Grigorova et al. [2007] an adaptive approach uses the user's feedback for suitable feature weight assignment and dynamic feature selection based on a set of replacement rules. Dynamic feature selection is also incorporated in the random forest-based approach of Wu and Zhang [2004b], utilizing the notion of balanced information gain to select the most optimal subset of features.

4.3 Exploring feature space

As was mentioned in the introduction, images of interest can be spread out over multiple areas in low-level feature space. For example in a feature space built up on color features, it is likely that images of differently colored tulips can be found in several parts of the space. Such a search can be performed using multiple query points [Jin and French 2003; Ortega-Binderberger and Mehrotra 2004], but exploring the feature space around each query point is often a slow process. Most interfaces only present a limited number of images to the user, putting a heavy burden on the user as navigating the space around each query point requires many iterations of feedback.

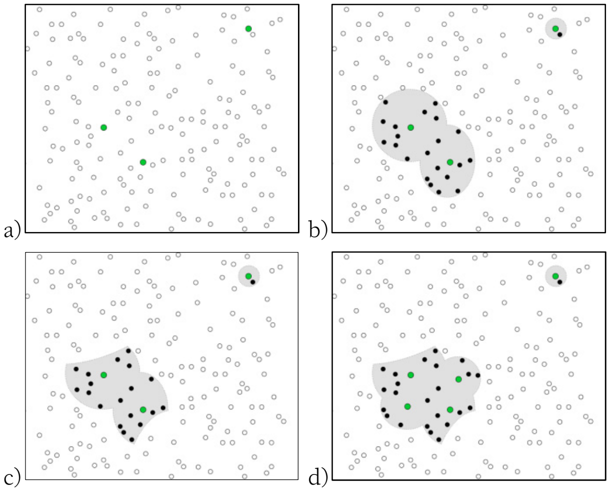


Figure 4.1: Establishing the search space.

To reduce user effort and allow efficient refinement of the relevant search space, we propose a novel technique where the search space surrounding relevant images is visualized and can be interactively adjusted by the user. In Figure 4.1 is illustrated how the user establishes the search space by exploring the nearest neighbors of relevant images. Initially, e.g. from a random selection of images from the database, the user marks one or more images as relevant (shown as larger dots with a green center in Figure 4.1a) and then for each of them proceeds to explore its nearest neighbors in feature space, increasing the number until on the border non-relevant images appear (the explored images are shown as black dots in Figure 4.1b). In subsequent steps, the search space can be refined by removing non-relevant images (Figure 4.1c) and exploring other relevant nearest neighbors (Figure 4.1d).

4.3.1 Interactive visualization

In retrieval systems based on relevance feedback, the user indicates her preferences regarding the presented results by selecting images as positive and negative examples. Subsequently, these feedback samples determine a relevance ranking on the image collection, and new images are presented to the user for the next round of feedback. In this paper, we propose to integrate feedback selection with a visualization mechanism that allows the user to quickly explore the local feature space surrounding an example image. The interaction provides a better sense of the local structure of the database, and allows the user to center on examples that best capture the desired image qualities.

At the start of an exploration interaction only the selected image is displayed. Then, by adjusting the *exploration front*, more and more of its nearest neighbors are shown, see Figure 4.2. When the number of nearest neighbors becomes too large to be displayed in a comprehensible manner, a random selection is displayed. The user can invoke the *deep exploration* browsing mechanism when she encounters an image of interest within the exploration range, and transfer the focus to this image to continue the exploration. This provides the user the opportunity to easily reach other areas in feature space, see Figure 4.3. This can be done as many times as the user wants, jumping from one area in feature space to another. This technique is particularly useful when the search seems to be 'stuck' and cannot improve with the current collection of relevant images. Also, using deep exploration to move from an isolated relevant image to a more densely populated relevant area has direct benefits for the feedback analysis, e.g. the feature selection and weighting approach will perform better with the additional data.

At any stage, the user may decide to treat a centered image as a positive example. In that case all images within the exploration front will also be treated as positive examples, so the exploration range should ideally encompass a high fraction of images considered as relevant. When desired, non-relevant images can be removed by the user at a later stage. Similarly, a selected example can be treated as a negative example. In the end, the retrieval system collects the positive and negative example images corresponding to the selected exploration ranges, and performs the relevance feedback analysis of the next section to determine both (i) the images estimated to be most relevant, and (ii) the images estimated to be most informative, i.e. optimal for display in the next iteration of exploration and feedback.

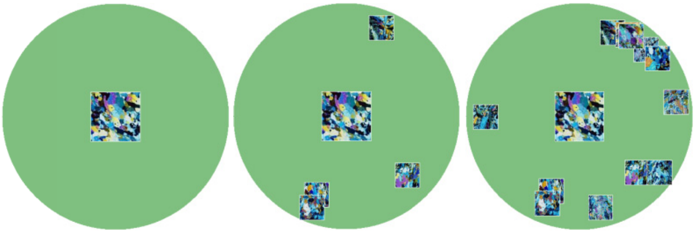


Figure 4.2: Exploring feature space: Initially only the selected image is shown in the center (left). The user expands the exploration range several times by scrolling the mouse wheel (middle, right). The distance of an image to the center increases linearly with their distance in feature space. The feature weights used for calculating the distance are discussed in Section 4.4.

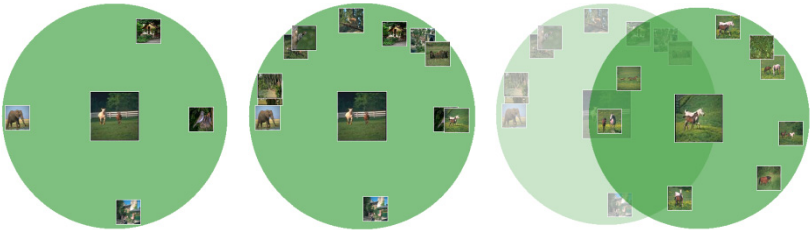


Figure 4.3: Deep exploration: The user is looking for images of horses and explores a relevant image to discover more. However, its nearest neighbors in feature space are not relevant at all (left). The exploration range is increased and a few relevant images are found (middle). The user moves the focus of exploration to one of them and this time many of its nearest neighbors are relevant (right).

4.3.2 Feedback sets

Let the positive feedback example set S_t^+ at iteration t consist of all selected relevant images gathered thus far

$$S_t^+ = \left\{(s_1^+, w_1^+, r_1^+), \cdots, (s_{n_t}^+, w_{n_t}^+, r_{n_t}^+)\right\} , \tag{4.1}$$

where, for each example image s_i^+, w_i^+ are the corresponding feature weights at the time of exploration and r_i^+ is the exploration range as selected by the user. The negative feedback example set S_t^- at iteration t is defined similarly. When a user re-explores an image, its previous feature weights and exploration range are updated with their new counterparts.

Let A_{ti}^+ be the set of images at iteration t within the exploration range of a positive feedback example

$$A_{ti}^+ = \left\{x \in D \middle| d_{w_i^+}(s_i^+, x) < r_i^+\right\} , \tag{4.2}$$

where x is an image from the image database and (s_i^+, w_i^+, r_i^+) are from the i-th tuple of S_t^+. Let A_{ti}^- at iteration t be defined similarly. We now define the active set A_t as the set of images at iteration t that are in at least one of the positive sets and not in the negative sets

$$A_t = \bigcup_{i=1}^{n_t^+} A_{ti}^+ \setminus \bigcup_{i=1}^{n_t^-} A_{ti}^- \ . \tag{4.3}$$

Constructing the most informative set

To determine the most informative images, we first calculate the information score T_I of each active image a at iteration t as the minimum distance to their associated feedback examples

$$T_I(a) = \min_{(s,w,r) \in S_t^+} d_w\,(s, a) \ . \tag{4.4}$$

Note that an active image can be in the exploration range of several feedback images. Next, we pick the images with the highest information scores, thus maximizing the minimum distances. As a result, images on the border of our search space will obtain the highest information score.

Constructing the best image set

Besides the most informative images, which allow the user to continue exploring the feature space, we keep an image set that contains the best images thus far. For each active image a at iteration t we calculate a relevance score T_R that is dependent on the distance to its feedback point(s)

$$T_R(a) = \sum_{i=1}^{n_t^+} \frac{1_{\{x | d_{w_i^+}(s_i^+, x) < r_i^+\}}(a)}{(1 + \gamma d_{w_i^+}(s_i^+, a))} \ , \tag{4.5}$$

where $1_A(x)$ is an indicator function, indicating the membership of x in set A and γ a constant that quantifies the rate of relevance decrease when an active image comes nearer to the border of the exploration range.

4.4 Feature weighting

The collection of explored feedback images provides us with a convenient setup for *local* feature selection. In particular, it allows us to take into account prior feature density by giving higher weight to feature regions where images cluster unexpectedly. This is desirable given that for features to which the user is indifferent, clustering will naturally occur at the feature regions of high prior density. In our method, the influence of the latter kind of features is suppressed. The resulting local feature weights are used to measure image similarity, (i) to new images to be explored, and (ii) to example images s, through the distance functions $d_w(s, x)$ of

Equations 4.4 and 4.5. For (ii), feature weights w_{ij} are computed for each sample image s_i. In the following we suppress image index i from the notation. Our approach is to estimate the prior feature value density corresponding to the local clustering of examples at the image under study and set the distance function weights accordingly.

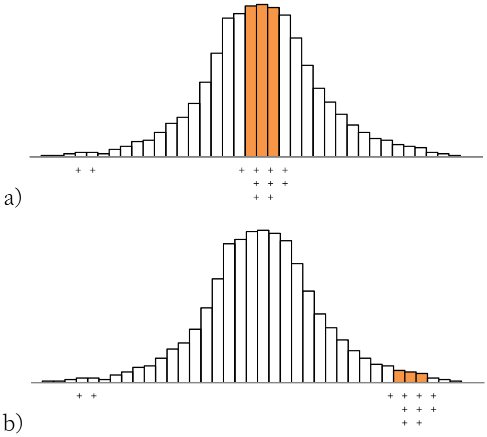


Figure 4.4: The local feature weight depends on the prior feature value density of the estimated interval.

In Figure 4.4 we have illustrated the weighting by prior density principle for a certain feature. The feature values of the collection of relevant feedback images are indicated by '+' symbols in both figures. In this example the images have formed two clusters in the feature value range, one small cluster and a larger one. In Figure 4.4a we see that the large cluster is located around the bulk of the feature density, whereas in Figure 4.4b it is located in a less dense region. We want to determine the optimal feature weight for a particular image. Assume that it has a feature value that falls within the large cluster. We establish a suitable feature interval, indicated in both figures by the orange bars, which we use to estimate the prior feature value density. The estimation technique is described below in more detail. Because the density contained by the interval is high in Figure 4.4a, it means that feature values in this interval are not very remarkable and consequently its weight should be low. In Figure 4.4b however, the density is low and therefore there is strong evidential support that the clustering around this particular image was intended by the user and consequently the associated feature weight should be high.

We use the distribution of the relevant examples to establish the feature interval that we should look at for estimating the prior feature value density. Consider the absolute deviations in feature j between image s and each of the active images

$$a \in A_t : \left| s_j - a_j \right| \ . \tag{4.6}$$

Taking the median of this sequence gives us the median absolute deviation, mad_j, which offers a measure of the spread of the active images around s for each of the features j. We will now let the local feature weight depend on the prior feature value density of the estimated interval $[s_j - \mathrm{mad}_j, s_j + \mathrm{mad}_j]$.

The density p can be estimated using standard non-parametric methods based on quantization. Since we have normalized our data, we found that a reasonable and fast approximation of this density can be obtained by means of the standard normal cumulative distribution function Φ

$$\begin{aligned} p_j &= p([s_j - \mathrm{mad}_j, s_j + \mathrm{mad}_j]) \\ &= \Phi(s_j + \mathrm{mad}_j) - \Phi(s_j - \mathrm{mad}_j)\,. \end{aligned} \tag{4.7}$$

We then transform this density into a weight w_j using

$$w_j = \frac{1}{1 + \beta p_j}\,, \tag{4.8}$$

where β is a constant that controls how fast the weight decreases for increasing density. High feature weights are achieved for intervals of small prior probability and low weights for intervals with high prior probability. Stronger selection can be enforced by first thresholding p_j i.e. setting p_j to zero when p_j is larger than the threshold. After weight normalization, the resulting distance function to image s is

$$d_w(s,x) = \sqrt{\sum_{j=1}^{M} w_j (s_j - x_j)^2}\,. \tag{4.9}$$

As an illustration, in Figure 4.5a we can see an image of a sunset that is explored with default feature weights. After several iterations of feedback the feature weights have changed and when the image is re-explored, as is shown in Figure 4.5b, its nearest neighbors are more relevant than they were before.

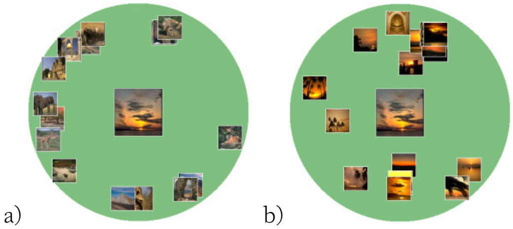


Figure 4.5: Default feature weights (a) vs. optimized weights (b).

4.5 Experiments

We used two image collections for the experiments, the Ponce texture collection [Lazebnik et al. 2005] and the Corel5k collection [Duygulu et al. 2002], where the images are categorized into 25 and 50 classes, respectively. All images are represented by the MPEG-7 homogeneous texture descriptor [Ro et al. 2001] and by a color histogram, based on a uniform quantization in 152 bins.

We have performed our experiments on four systems, i) one using our proposed exploration interface, deep exploration and feature weighting ('Deep Explore'), ii) one using the exploration interface without deep exploration, but with feature weighting (Feature Explore'), iii) one using the exploration interface without deep exploration and feature weighting (Standard Explore'), and iv) one using a standard interface and the query point movement technique as proposed by Rocchio ('Rocchio'). Because one of the strengths of our approach is the interface that provides easy access to additional relevant and non-relevant images, we acknowledge that the Rocchio system cannot be fairly compared with both Explore systems. However, since the systems try to achieve the same goal and are given the same data to work with, we believe that in this sense all systems are comparable.

For each of the image classes we have set up experiments, where the goal is to find all images belonging to that class within at most 20 iterations. Every iteration the user is presented with 40 images. For all Explore systems, these images are composed of the most informative images as calculated by Equation 4.4. For the Rocchio system, these images consist of the resulting images after performing query point movement. Besides the images on which feedback can be given, a separate result set is kept that contains the best ranking images. For the Explore systems, these images are composed of the top images as calculated by Equation 4.5. For the Rocchio system, the best ranking set is the same set as the informative set, containing the resulting images after query point movement. In our results we define *precision* as the number of relevant images found over the top 40 best ranking images, and *recall* as the number of relevant images found thus far over the total number of existing relevant images, which is 120 per Ponce texture class and 100 per Corel5k class.

Because we did not have access to a large group of users to participate in the experiments, we have simulated realistic user behavior to the best of our ability. Since real users generally do not want to give much feedback, in our simulations per iteration a maximum of 5 images are marked as relevant and a maximum of 5 as non-relevant. For the Deep Explore system, the simulated user will attempt to shift the focus of exploration when the number of relevant nearest neighbors of an explored image is small. In this situation, this user will increase the exploration range to only the 100 nearest neighbors and, when one or more relevant images are contained within the exploration range, the one that is furthest away will be explored. Up to 4 shifts of exploration can be chained together and the image with the largest exploration range is used as a positive feedback image. For each experiment we fill the initial screen with random images. Retrieval performance is dependent on which images appear within this initial screen, specifically on how many of these random images belong to the class of interest. This issue is commonly referred to as the page zero problem [La Cascia et al. 1998]. Therefore we perform the experiment for each class 100 times and average the results. If the initial screen does not contain any relevant image, we generate a new set of random images until at least one relevant image is shown.

As we can see in Figure 4.6, the average precision on the Ponce image database rapidly increases for all Explore systems. We notice that using feature weighting gives an increase in performance, and by additionally using the deep explore technique the results improve even further. As a comparison, the Standard Explore system reaches 80% precision after 4 iterations, the Feature Explore system after 3 iterations and the Deep Explore system already after 2 iterations. Even after the first few iterations all Explore systems keep improving. The Rocchio system finds almost all the relevant images it is able find after the first few iteration and hardly improves after that. It reaches a maximum precision of just over 55%. We can also see that all Explore systems manage to discover between 65% and 70% of all relevant Ponce images per class, with the Deep Explore system finding the most. Rocchio on the other hand has just over 30% recall. On average, the accuracy of the Explore system improves considerably over the Rocchio system after the first iteration.

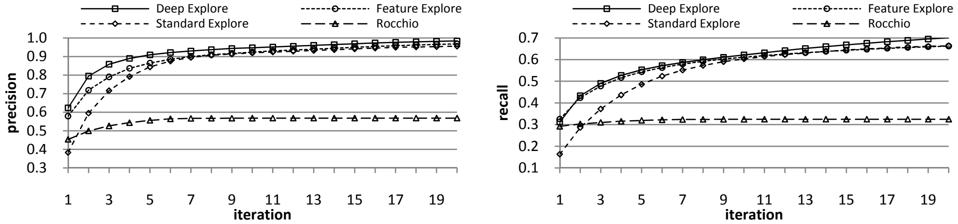


Figure 4.6: Average precision (left) and recall (right) results on the Ponce collection.

In contrast with the uniformity of the classes in the Ponce database, the classes in the Corel5k are much more diverse, and this results in lower overall performance for all systems. One of the strengths of the Explore systems – collecting relevant images instead of ranking all database images – also becomes one of its weaknesses if not many of the nearest neighbors are relevant. During our experiments we observed that it relatively frequently happens that the nearest images to a relevant image in the Corel5k database are not relevant at all. Another issue that has an impact on the results of all our systems is that in the Corel5k database the same type of images appear in several categories, for instance polar bears appear in the categories ‘alaskan wildlife’, ‘bears’ and ‘polar bears’. We can clearly see these issues affect the recall results, shown in Figure 4.7, since the highest achievable rate for the best performing system is only 35%. Nonetheless, this dataset is widely used in the research community and is therefore an interesting benchmark for our experiments. When we look at the precision results we notice that using feature weighting improves performance only slightly over not using feature weighting, which is a natural result of the lack of discriminatory power of color and texture features for predicting the Corel categories. Especially in this situation the deep exploration technique helps to boost the performance of the Deep Explore system significantly over both the Feature Explore and Standard Explore systems, because it is able to navigate to other areas in feature space where larger clusters of relevant images can be found.

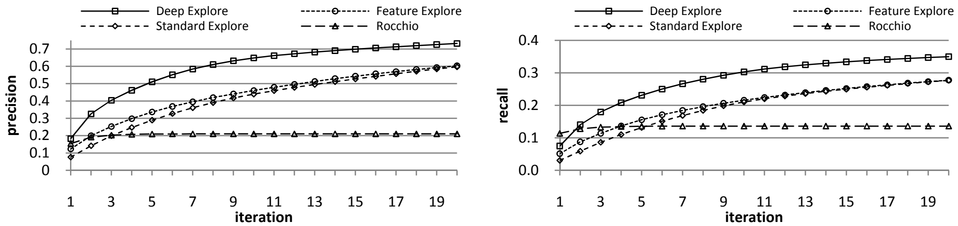


Figure 4.7: Average precision (left) and recall (right) results on the Corel5k collection.

4.6 Conclusions

In this chapter, we have discussed an exploration-based interface that allows the user to visually and interactively explore the feature space around images. Using the deep exploration technique, relevant areas in feature space can be discovered that would otherwise remain unnoticed. In addition, when an image is explored, its optimal set of feature weights is automatically determined using all images contained within the relevant regions, based on the evidential support for the relevance of each single feature. We performed user experiments on two well-known image databases and the results indicate that the new deep exploration approach coupled with the optimal feature weighting technique lead to an improvement of the number of relevant images collected.

5. TOP-SURF: a visual words toolkit

TOP-SURF is an image descriptor that combines interest points with visual words, resulting in a high performance yet compact descriptor that is designed with a wide range of content-based image retrieval applications in mind. TOP-SURF offers the flexibility to vary descriptor size and supports very fast image matching. In addition to the source code for the visual word extraction and comparisons, we also provide a high level API and very large pre-computed codebooks targeting web image content for both research and teaching purposes.

5.1 Introduction

In our world vision plays a very important role and computers are slowly catching up with the qualities of human vision. In the early days image descriptors were based on low-level features, such as colors and edges, but nowadays the descriptors are approaching image analysis from a higher level, resulting in image descriptors that are based on, for instance, salient details or image patches. Interest points are a specific kind of salient details, which describe locations in an image that are 'interesting' in a certain way. In this chapter we present TOP-SURF, which is an image descriptor that combines interest points with visual words. It harnesses the high-level qualities of interest points, while significantly reducing the memory needed to represent and compare images. Our visual word dictionaries (codebooks) are created by analyzing the interest points extracted from several millions of web images. The TOP-SURF descriptor is completely open source, which includes the libraries it depends on. Furthermore, the source code can be easily compiled and included in an existing project, or can be used in binary form where its functionality is available through an accessible API.

Because TOP-SURF is based on SURF [Bay et al. 2008], we will first shortly introduce this descriptor in Section 5.2, before discussing our descriptor in more detail in Section 5.3. Along the way we illustrate the differences in descriptor size, description time and matching time between both descriptors. We also compare their performance using a near-duplicate detection scenario as a showcase. Finally, in Section 5.4 we will describe the TOP-SURF API, open source licenses, documentation and other possible scenarios in which our descriptor would be useful.

5.2 SURF

SURF is one of the best interest point detectors and descriptors currently available. It has been shown to outperform the other well-known methods based on interest points SIFT [Lowe 2004] and GLOH [Mikolajczyk and Schmidt 2005].

5.2.1 Representing an image

The SURF technique uses a Hessian matrix-based measure for the detection of interest points and a distribution of Haar wavelet responses within the interest point neighborhood as descriptor. An image is analyzed at several scales, so interest points can be extracted from both global ('coarse') and local ('fine') image details. Additionally, the dominant orientation of each of the interest points is determined to support rotation-invariant matching. An example image and its detected interest points are shown in Figure 5.1.

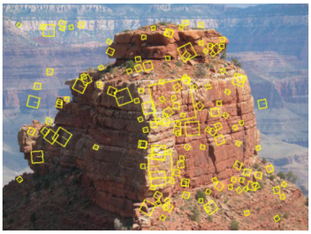


Figure 5.1: Detected interest points in an image, including their orientation and scale.

To determine the average number of extracted interest points per image, we used a collection of 100,000 images downloaded from the internet, which included logos, graphics, celebrity shots, stock photography and travel-related imagery. These images represent well what one would generally encounter when browsing on the internet, with the exception of small icons and banners that have been left out. These images have dimensions ranging between 83 and 640 pixels, with an average size of 460x400. Yet, for extracting the interest points we resized all images to 256x256.

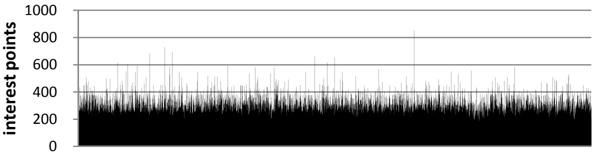


Figure 5.2: Number of interest points detected in the first 25,000 images of the collection.

We show the number of detected interest points for a selection of the images in Figure 5.2. The number of interest points found in an image ranged between 0 and 1057, and was on average 176 with a standard deviation of 85,3. It is thus certainly possible that no interest point is detected in an image at all, which can for instance occur when an image has a single color, although this does not happen frequently. Because each interest point is associated with a 64-element descriptor, the total descriptor size in our image sets thus ranges between 0KB and 270KB per image, with an average of roughly 45KB. On average 0.37s was required to extract the interest points from an image. The results were obtained using standard dual-core 2.4GHz workstation equipped with 3GB RAM. Note that more or less interest points can be detected when the images are resized to a different resolution than 256x256.

5.2.2 Matching an image

When enough interest points in the first image match those in the second image, the images are likely to depict the same scene or object(s). To determine these matches the authors of SURF use the *nearest neighbor ratio* matching technique [Lowe 2004], which only accepts highly confident matches between points. For each point x in the first image, we find both the best matching point y_1 and the second best matching point y_2 in the second image, where $y_1 \neq y_2$. Points are compared by first ensuring the sign of the Laplacian of their descriptors correspond and, if so, calculating the Euclidean distance between the descriptors. The match $x \sim y_1$ is then accepted only if the ratio $r = |x - y_1|/|x - y_2|$ is lower than a certain threshold $\lambda \in [0,1]$. Intuitively, if $\lambda = 1$ no filtering is performed, while more and more matches are rejected as λ decreases towards zero. The SURF authors used a threshold $\lambda = 0.65$, which is the threshold we used as well. The total number of confident matches can then be compared with a second threshold to detect whether or not the second image is similar to the first image. An alternative approach would be to rank all images in the database by the number of confident matches, and the highest ranking image can then be considered as the most similar one to the first image.

We used a separate set of query images, consisting of 100 travel-related photos, and matched them with all the 100,000 images. On average 12ms was required to match one query image with all the other images. We performed the matching on a quad-core 2.4GHz workstation equipped with 8GB RAM in order to handle the large descriptor size.

To determine the accuracy of the SURF descriptor, we created several near-duplicates of each of the 100 photos. These copies were compressed, scaled, framed, colorized or overlaid with text and a logo, see for example Figure 5.3. The exact copies we used are discussed in more detail in Chapter 6, in which we perform extensive experiments using multiple near-duplicate detection algorithms, including our TOP-SURF descriptor. For our current experiment, we embedded the duplicate images in the collection of 100,000 images. When matching the original images with all other images in the collection, ideally the copies will be ranked before all other images. We used *mean average precision* (MAP) as the evaluation measure, which is calculated by first determining the average precision over all copies for each of the queries and then averaging these average precision values. For clarity, in our results we define *precision* as the number of copies found over the total number of images looked at. Our evaluation found that the MAP for SURF was 0.31.

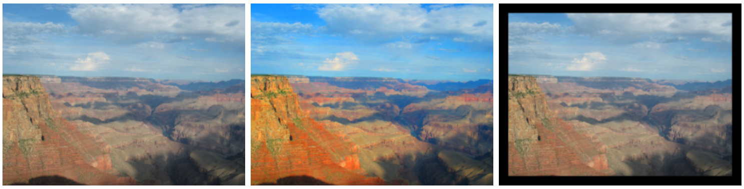


Figure 5.3: Examples of near-duplicate images: original image (left), increase in saturation (middle), framing (right).

5.3 TOP-SURF

When considering to find matches in collections containing millions of images it is clear that using the SURF method in its default form is storage-wise infeasible. One of our reasons for developing TOP-SURF was to overcome this issue by significantly reducing the descriptor size.

5.3.1 Representing an image

Several steps need to be performed in order to calculate the TOP-SURF descriptor of an image.

5.3.1.1 Representative interest points

We used a large set of diverse training images consisting of 1 million images downloaded from the internet, 1 million images downloaded from Flickr and 3000 land- and cityscape photos from our personal collections. Our aim was to compose a general purpose web imagery set that would be representative for the kind of images used by researchers and students in content-based image retrieval.

For each of these images we extracted their SURF interest points and randomly selected 25 points. Because some of the images did not have much detail, it occasionally occurred that less than 25 points were extracted and in those situations we used all of them. Due to limited amount of memory available we could not use all extracted points, and eventually settled on a collection containing 33.5 million interest points. The time required to collect all these points was 120 hours.

5.3.1.2 Clustering into visual words

We devised an approach based on the bag-of-words technique of Philbin et al. [2007] to group the collection of representative interest points into a number of clusters. Since each interest point can be considered a location in a 64-dimensional space, we can see this process as analyzing the locations of all 33.5 million interest points and gathering them into a certain number of groups. First, to find an initial location for each cluster we randomly and uniquely assigned it the location of one of the interest points. Then for each cluster we determined its 100 nearest neighbors, i.e. its closest interest points. If a point was close to multiple clusters we only assigned it to the cluster it was closest to. We then updated each cluster to become the average of its current location and that of its nearest neighbors. To ensure stability of each of the clusters we performed this process 1000 times.

Because discovering the exact nearest neighbors in such a high-dimensional space is quite time consuming, we used an approximate nearest neighbors technique based on a forest of randomized kd-trees [Muja and Lowe 2009] to speed up this process. A regular kd-tree is a binary search tree in which each node represents a partition of the k-dimensional space. The root node represents the entire space, and the leaf nodes represent subspaces containing non-overlapping small areas of the space. At any node, only one of the k dimensions is used to partition the space, which generally is the dimension that has the highest variance for all its associated points. The value that is used to partition the space can then be the median or mean along that dimension. In contrast, in the randomized kd-tree the partitioning dimension is randomly chosen from a set of the dimensions with highest variance and the partitioning value is randomly chosen using a point close to the median. By considering all these randomized kd-trees together an overlapping partitioning of the space is obtained, which reduces the chance of incorrectly assigning nearest neighbors to points that fall close to a partition boundary.

Depending on the intended usage of our descriptor only a small number of clusters may be necessary, whereas in other instances a large number may be required. Therefore we clustered the interest points several times, choosing a different number of clusters that ranged from 10,000 to 500,000. The clustering process was done on a high-performance blade server and required 28GB RAM. In Figure 5.4 we show the time needed to cluster all these points into the varying numbers of clusters.

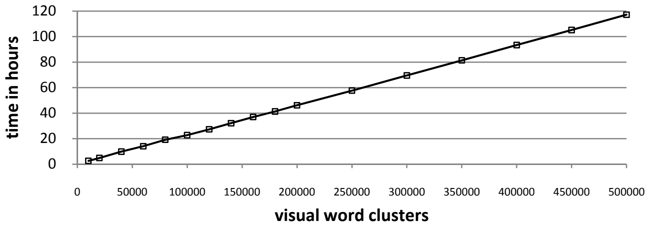


Figure 5.4: Required time to cluster the collection of representative interest points into visual words.

The final clusters are commonly referred to as the *visual word dictionary*. Similar to a document consisting of textual words, an image can be interpreted as consisting of visual words. Since in a collection of documents some words appear more frequently than others it is likely that in a collection of images some visual words appear more often than others as well. In our situation, the aim is to emphasize the visual words that do not occur frequently, because they can be considered to be more special (descriptive) when they are found in an image. To assign the visual words weights for emphasis we incorporated a tf-idf weighting technique [Salton and McGill 1983]. Tf-idf weighting combines the principle of term frequency, i.e. how often a particular term appears in a document, with the principle of inverse document frequency, i.e. how infrequently documents contain this particular term. If a document contains a particular term that is quite rare (e.g. the word 'diamond'), it is considered to be more special and thus will receive more emphasis than when the particular term is quite common (e.g. the word 'the'). Assume there are N documents in the collection, and that term t_i occurs in n_i of them, then

$$idf(t_i) = log\frac{N}{N_i} \ . \tag{5.1}$$

Furthermore, given a particular document d_j that contains n times term t_i and in total contains k terms, term frequency is defined as

$$tf(t_i, d_j) = \frac{n}{k} \ . \tag{5.2}$$

The final score a certain term t_i receives for occurring in a certain document d_j is then defined as

$$score(t_i, d_j) = tf(t_i)\ idf(t_i, d_j) \ . \tag{5.3}$$

Following the same analogy as mentioned before, in our situation 'term' thus refers to 'visual word' and 'document' to 'image'. Our idf-weights were obtained by recalculating all interest points of a subset of the training images and analyzing which of the visual words they would be associated with.

5.3.1.3 Selecting the most descriptive visual words

Given a particular total number of available visual words, we can now calculate the TOP-SURF descriptor of an image. First we extract its regular SURF descriptor. We then convert the detected points into a frequency histogram of occurring visual words, by analyzing which visual word each interest point is most similar to. Next, we apply the tf-idf weighting to assign a score to all the visual words in the histogram. To form our image descriptor we finally select the highest scoring visual words. Because we only select the top N visual words we thus named the descriptor TOP-SURF. An illustration is shown in Figure 5.5. Note that our descriptor only requires 8 bytes per selected visual word. Storing a collection of 100,000 images would roughly require 4.5GB when using the SURF descriptor, however this would only require 80MB with TOP-SURF when keeping the top 100 visual words, which is a reduction of more than 50 times.

Like we did with SURF in the previous section, we used the dual-core workstation – thus not the blade server – to calculate the TOP-SURF descriptors for all the 100,000 images. We did this (i) using the various dictionaries that contained 10,000-500,000 visual words and (ii) using different numbers of selected highest scoring visual words that ranged from 10-200 in steps of 10. The time that was required to extract a TOP-SURF descriptor is on average 0.44s. Since the descriptor includes the calculation of the SURF interest points, which required 0.37s as we observed before in Section 5.2.1, the conversion of the interest points to their visual words and the extraction of the top N points thus approximately took 0.07s. Note that the time needed to determine the top 10 visual words is the same as determining the top 200, because all visual words still have to be sorted on their scores. From the results we additionally observed that the time to extract our descriptor slightly increased from 0.42s to 0.46s as the dictionary got larger.

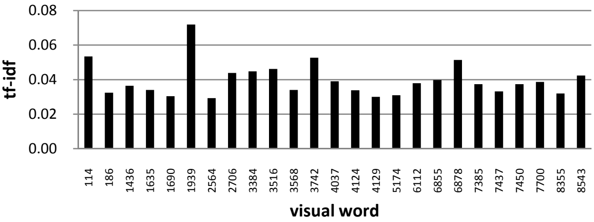


Figure 5.5: The histogram of the 25 highest scoring visual words of the image shown in Figure 5.1 when using a dictionary of 10,000 visual words.

5.3.2 Matching an image

To compare the TOP-SURF descriptors of two images we determine the normalized cosine similarity d_{cos} between their tf-idf histograms T_A and T_B

$$d_{cos} = 1 - \frac{T_A \cdot T_B}{|T_A|\ |T_B|}\ . \tag{5.4}$$

A distance of 0 means the descriptors are identical and a distance of 1 means they are completely different. Note that, by definition, comparisons with an image in which zero interest points have been detected will always result in a distance of 1, which is the desired behavior. To determine the matching time between TOP-SURF descriptors, we used the same set of query images as before when we matched SURF descriptors. On average 0.2ms was needed to match one query image with all 100,000 test images. In comparison with SURF this is very fast, since only a small number of visual words need to be compared. In contrast, with SURF each interest point of a query image needs to be compared to the interest points of all other images, requiring much more time. We noticed that matching was slightly faster with descriptors that used only a small number of selected visual words. Matching was also faster as the dictionary size increased, because in this situation it is less likely for the visual words in two descriptors to exactly match, in which case these can be skipped and thus do not require further analysis.

We performed the same near-duplicate image detection experiment as with SURF and our results are shown in Figure 5.6 for various dictionary sizes. We can see that a larger dictionary yields a higher accuracy for small numbers of retained visual words. In this experiment, the dictionaries containing 100,000 visual words and up gave virtually the same results. As the number of retained words increases, the performance goes up for all dictionaries and levels out at around a MAP of 0.96. Note that the TOP-SURF descriptor size is dependent on the number of visual words retained and not on the dictionary size.

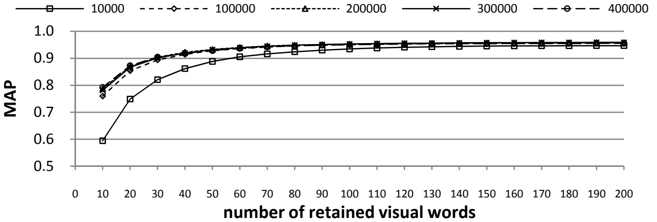


Figure 5.6: Mean average precision for various dictionary sizes varying from 10,000 to 400,000.

Overall, we can observe that the TOP-SURF descriptor significantly outperforms the SURF descriptor when it comes to retrieval accuracy, descriptor size and matching time in the context of near-duplicate image detection.

5.4 Source code

The TOP-SURF descriptor is completely open source, although the libraries it depends on use different licenses. Because the original SURF descriptor is closed source, we used the open source alternative called OpenSURF [Evans 2009], which is released under the GNU GPL version 3 license. OpenSURF itself is dependent on OpenCV [Bradski 2000] that is released under the BSD license. Furthermore we used FLANN [Muja and Lowe 2009] for approximate nearest neighbor matching, which is also released under the BSD license. To represent images we used CxImage (*www.xdp.it/cximage.htm*), which is released under the zlib license. Our own source code is released under a combination of the GNU GLP version 3 license and the Creative Commons Attribution version 3 license. The latter license simply asks anyone who uses our library to give us credit. All these licenses are compatible with each other.

The source code of our descriptor can be obtained from the TOP-SURF webpage at *press.liacs.nl/researchdownloads/topsurf*. On this page we have also posted documentation and instructions on how to include the library in your own projects. We additionally have provided binaries, samples and a graphical user interface. Because our visual word dictionaries can be quite large, they are offered as separate downloads. All our deliverables are currently only offered for the Microsoft Windows platform, both for 32- and 64-bit systems. The source code is presented in a Microsoft Visual Studio 2008 C++ project, although there is no reason to believe it cannot be converted to a project from earlier or later versions of Visual Studio. The code includes all the source code of the libraries it depends on for easy compilation.

5.4.1 API

Our descriptor API offers the following functions:

- TopSurf_Initialize
  Initialize the library.
- TopSurf_Terminate
  Terminate the library.
- TopSurf_LoadDictionary
  Tell the library which visual words dictionary to use.
- TopSurf_CreateDictionary
  Create a completely new visual words dictionary.
- TopSurf_SaveDictionary
  Save a newly created dictionary to disk.
- TopSurf_ExtractDescriptor
  Extract the descriptor of an image.
- TopSurf_VisualizeDescriptor
  Display the locations of the detected visual words.
- TopSurf_CompareDescriptors
  Compare two descriptors and return the distance between them.
- TopSurf_LoadDescriptor
  Load a descriptor from disk.
- TopSurf_SaveDescriptor
  Save a descriptor to disk.
- TopSurf_ReleaseDescriptor
  Release the memory used by a descriptor.

The API only has to be used when accessing the TOP-SURF descriptor through a DLL. When the source code is added to a project there is naturally more control and freedom, since functions can be called directly.

5.4.2 Benefits and uses

For convenience, we allow the user to request all detected visual words in an image and not just the top few. In addition, we extended our descriptor to also include the locations where their original interest points were detected in the image. Both these options allow our descriptor to be used in a variety of situations. For example, an application can analyze the co-occurrence of particular visual words within an image and combine visual words into *visual phrases*, opening up possibilities for improved matching of objects and people. Because of its fast matching speed and low memory requirement the TOP-SURF descriptor is especially useful for mobile and embedded applications, since the devices they will run on are generally restricted by processing power and memory.

Because our descriptor is easy to use and straightforward to integrate into projects, it is not only beneficial to researchers in the content-based image retrieval community, but also very suitable for use in student projects. Examples of student research projects at our computer science department are developing new visual phrase and visual theme search methods based on the pre-computed dictionaries, and automatic robotic navigation based on real time video input.

5.5 Conclusions

TOP-SURF is a high-performance image descriptor that can be used in a wide range of applications. It is not only very compact, i.e. using little memory, but also exhibits fast matching. Because the descriptor is completely open source, it has all the benefits that open source software provides, such as the freedom to modify and redistribute the code. In addition, we provide pre-computed visual word codebooks, making it easy to start using the descriptor.

6. Near-duplicate image detection

Digital information is distributed in large quantities across the world and often more or less the same content can be found at multiple locations. In this chapter we focus on imagery available on the internet and evaluate in which ways near-duplicate images differ from each other. We provide a comparative study of content-based near-duplicate image detection methods and specifically target their performance in relation to their descriptor size, description time and matching time to assess their feasibility of application to large image collections (> 1 million). The evaluated methods include research literature methods based on interest points matching, discrete cosine and wavelet transforms, color histograms and biologically motivated visual matching.

6.1 Introduction

The amount of media items produced on a daily basis is truly immense and many of these items are distributed to all corners of the world to be published by growing numbers of digital media outlets. This is not only the case for news stories, where often the same video footage is shown on multiple television channels and the same text appears in several newspapers, but this also applies to other sources of information such as photographic imagery. In other words, there is significant redundancy in the information that is available to the public. With current estimates putting the number of images on the internet into the tens of billions, it is not difficult to imagine that many of these are in fact near-duplicate versions of each other. A study is presented in Foo et al. [2007a] that investigates the extent of this issue for a diverse selection of popular search terms and the authors identify two important factors for predicting if a search term is likely to result in image rankings with many duplicates or near-duplicates. First, images on certain topics are relatively rare and this results in the few available images to be reused on many different sites. Second, images are reused often because of their popular content. Near-duplicate images can be found in many contexts, such as being integrated into web page design, desktop backgrounds and advertisements. Possible uses for detecting copies include introducing more diverse content to image search results by aggregating (near-)duplicate images, and identifying instances of copyrighted material.

Our primary goal is to evaluate which methods are the best candidates for near-duplicate image detection on the world wide web. Copy detection techniques that are based on adding hidden information to the original content in order to easily discover its copies, e.g. watermarking [Cox et al. 1997], require having complete control over the originals. However, in most situations the originals are already available to the public without the hidden information, rendering these techniques useless. Additionally, such methods cannot be applied to photographs taken of public objects and scenes, e.g. photos of the Mona Lisa painting, when they are so similar that they can be considered to be duplicates. In this chapter we present an evaluation of several near-duplicate image detection methods from the research literature [Kim 2003; Chang et al. 1998; Sebe and Lew 2001; Bay et al. 2008]. The authors typically provide some basic benchmarking, but rarely compare their results to other methods, and none of these studies use test sets sizes comparable to ours. We use a large set of originals and copies (altered versions of the original by a diverse set of realistic transformations) and embed them in two collections of one million images each. The first collection consists of images downloaded from all over the internet, and the second collection consists purely of images from the popular Flickr photo sharing website.

We evaluate the methods by two important, yet potentially contradicting, criteria. First by accuracy, typically measured in terms of false positive and false negative rates or their close counterparts precision and recall. Second by computational requirements, i.e. by measuring indicators for usage of main memory, hard disk storage, and processing times for image description and image matching. We are especially concerned with the scalability of the detection methods with respect to these measurements. In Foo et al. [2007a] an exploratory study is presented comparing methods targeted at near-duplicate detection of web images. However, the focus of that paper is to detect copies in the results returned by search engines. Because we are aiming to detect duplicates in all indexed images, we need to assess feasibility at much larger scales. Many studies have focused on test sets with a size in the range of 10,000 to 40,000 images [Chang et al. 1998; Lu and Hsu 2005; Ke et al. 2004; Foo et al. 2007a, 2007c; Foo and Sinha 2007b], but in the context of web search there is clearly a need to use larger test sets. Such kind of studies are not frequently carried out, exceptions being Wang et al. [2006] with 1.4 million and Ghosh et al. [2007] with 10 million web images. During the past ten years the content-based retrieval community has been increasing the size of the credible test sets, from hundreds to thousands. It has repeatedly been found that methods that work very well on a thousand images frequently perform poorly on ten thousand images. Our motivation for using databases containing one million images is thus to show scalability for the future.

The structure of this chapter is as follows. In Section 6.2 we describe the near-duplicate detection methods we have compared in this study and in Section 6.3 we present the image collections. The experimental setup and the obtained results are presented in Section 6.4. We conclude in Section 6.5.

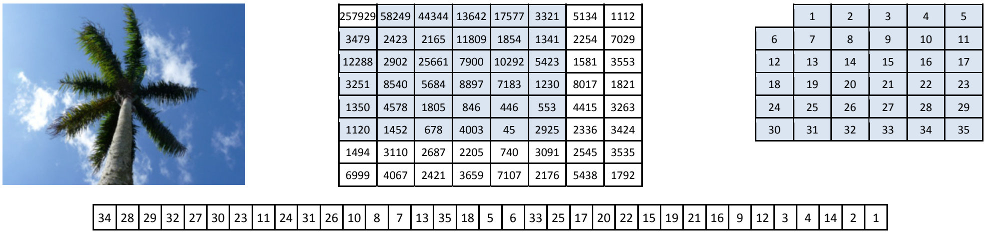
6.2 Near-duplicate image detection methods

From the research literature we have selected four well-known and representative near-duplicate image detection methods. Each of them uses a different representation as basis for detecting copies, namely discrete cosine transform, discrete wavelet transform, color histograms and interest points. We have implemented these four methods to the best of our ability, based on the sequence of steps and values used as described in their respective papers. In addition, we have developed three other methods ourselves of which the first is based on intensity differences, the second on human vision and the third is a hybrid of both.

In a real-world setting appropriate distance cut-off values need to be established for the methods in order to differentiate between copies and non-copies. However, for the evaluations performed in this paper we used the ranking of distances between images, allowing us to obtain meaningful accuracy and performance results by imposing varying distance thresholds

6.2.1 Discrete cosine transform

For the representative method using the discrete cosine transform (DCT), we used the algorithm of Kim [2003]. The images are first converted to grayscale and then resampled using intensity averaging to an 8x8 size. The resulting 64 intensities are transformed into a series of coefficients by performing a 2D DCT, of which the low-frequency AC coefficients are stored in a rank matrix. Duplicates of an image can be detected by comparing and thresholding the L_1 distance between the rank matrices. The authors found experimentally that using only the 35 low-frequency AC coefficients, which are located in the upper-left of the DCT coefficient matrix, offered the best ability to discriminate between copies and non-copies. An illustration is shown in Figure 6.1.



257929	58249	44344	13642	17577	3321	5134	1112
3479	2423	2165	11809	1854	1341	2254	7029
12288	2902	25661	7900	10292	5423	1581	3553
3251	8540	5684	8897	7183	1230	8017	1821
1350	4578	1805	846	446	553	4415	3263
1120	1452	678	4003	45	2925	2336	3424
1494	3110	2687	2205	740	3091	2545	3535
6999	4067	2421	3659	7107	2176	5438	1792

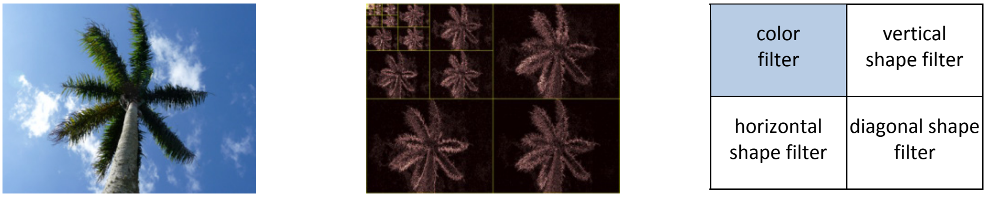
	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31	32	33	34	35

34	28	29	32	27	30	23	11	24	31	26	10	8	7	13	35	18	5	6	33	25	17	20	22	15	19	21	16	9	12	3	4	14	2	1

Figure 6.1: Example of ranking 35 DCT coefficients: From the input image (top-left) an 8x8 DCT coefficient matrix is created (top-middle), where the 35 coefficients to be selected are highlighted. The indices of these coefficients (top-right) are ranked (bottom) based on the coefficient magnitudes.

6.2.2 Discrete wavelet transform

In the work by Chang et al. [1998] each image is resampled to 256x256 pixels and then converted to a human perceptual color model. For each of the three color channels, Daubechies' discrete wavelet transform (DWT) is used multiple times to reduce the number of coefficients. The resulting 8x8 low frequency coefficients are used as color filter, while the 8x8 horizontal, vertical and diagonal high-frequency coefficients are separately summed and thresholded and used as shape filter, see Figure 6.2. Following the original work, these thresholds were determined using a training set of query, web and flickr images. Images are compared by first performing a filtering step to ensure that the values of the shape filter are identical, and then by applying fine-grained matching that looks at the L_2 distances between the color filters. The resulting duplicates of an image are those images that pass the filtering step and have an overall distance smaller than a given threshold.



color filter	vertical shape filter
horizontal shape filter	diagonal shape filter

Figure 6.2: Decomposition of an image (left) into its low frequency and high frequency wavelet components (middle). The coefficients in the upper-left quadrant are used to compute the image descriptor (right). This is done for each color channel.

6.2.3 Color histograms

The representative work for color histograms by Sebe and Lew [2001] begins by converting each image to the HSV color space and creating a quantized color histogram with 16 bins for hue, 4 bins for saturation and 4 for value. Using a training set of originals and copies, the absolute differences between their histograms are modeled into a probability mass function (pmf). To determine whether an image is a duplicate of a query image first the normalized histogram of the absolute bin differences (ndh) between their color histograms is calculated, see Figure 6.3 for an illustration. Then the L_1 distance is computed between the pmf and the ndh. If the distance is smaller than a certain threshold the images are considered to be copies of each other.

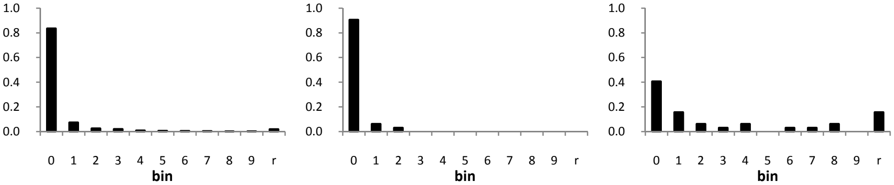


Figure 6.3: The pmf trained on originals and copies (left). The ndh between the color histograms of two copies (middle). The ndh between the color histograms of two non-copies (right). The first ten bins in the ndh model the small differences and the last bin is used for the remainder.

6.2.4 Interest points

Because SURF [Bay et al. 2008] has been shown to outperform the other well-known methods based on interest points SIFT [Lowe 2004] and GLOH [Mikolajczyk and Schmidt 2005], we have selected this method as the leading technique for the interest points representation. As we demonstrated in Chapter 5, it is infeasible storage-wise to use the SURF descriptor in its default for large image collections, because it roughly requires 45KB to represent an image, and we therefore developed our own descriptor called TOP-SURF. TOP-SURF is based on SURF, but reduces the amount of memory needed by more than 50 times. For the experiments in this chapter we used a dictionary of 200,000 visual words, which were obtained by clustering the detected interest points from a training set of query, web and flickr images using the bag-of-words technique proposed by Philbin et al. [2007]. The detected interest points in an image are then converted into a frequency histogram of occurring visual words. Next, we applied tf-idf weighting [Salton and McGill 1983] to select the most descriptive visual words from the frequency histogram as the image descriptor, see Figure 6.4 for an example. By applying this weighting and selection technique the method can compete with the other methods in terms of required bytes per image.

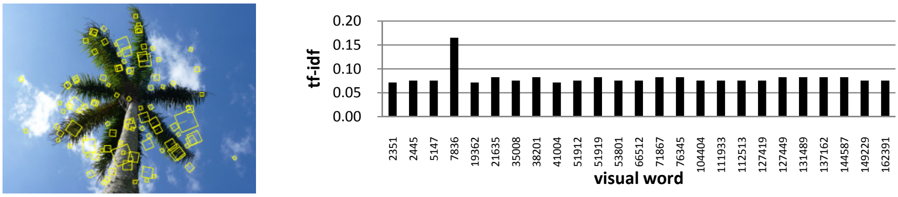


Figure 6.4: The detected interest points in an image (left) and its histogram of the 25 most descriptive visual words (right).

After preliminary experimentations with varying numbers of visual words, shown in Figure 6.5, we found that using the top 100 words gave a very high accuracy for an acceptable matching time and descriptor size. Duplicates of an image are found by determining the normalized cosine similarity between their tf-idf histograms and applying a threshold.

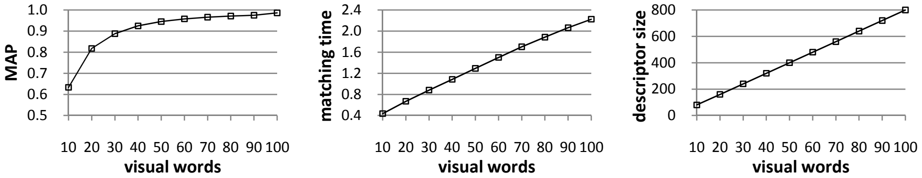
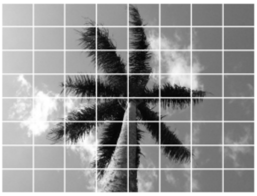


Figure 6.5: Accuracy in mean average precision (left), matching time per query image in seconds (middle) and descriptor size in bytes per image (right) vs. varying numbers of visual words.

6.2.5 Median

The first of our in-house developed methods is based on intensity differences. After converting each image to grayscale, the image is divided into 64 blocks (8 horizontal by 8 vertical). The mean intensity is calculated for each of these blocks and is compared with the median intensity of the entire image. The image descriptor consists of a bit vector flagging if block means are greater than the overall median, see Figure 6.6. Duplicates of an image are detected by calculating and thresholding the number of bit errors between vectors.



1	1	1	1	0	0	0	0
1	1	1	0	0	1	0	0
1	1	1	0	0	1	0	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	0

Figure 6.6: The image cut up in 64 blocks (left) and its bit vector shown as a matrix (right).

6.2.6 Retina

Human vision is a well-researched system, particularly in biology, that has many promising applications in computational visual understanding algorithms. This approach is based on results from the neuro-biology field described by Levine [1985], which reveal that the human retina has an exponentially decreasing density of cones in the eye with decreasing visual acuity as measured from the center of the retina. Our approach tries to roughly replicate this by placing blocks at pre-determined locations according to an exponential distribution, as is shown in Figure 6.7.

Preliminary experimentations indicated that using an image resolution of 256x256 and placing 50 blocks of size 5x5 pixels gave the best tradeoff between accuracy and descriptor size. In Figure 6.8 the results are shown for varying numbers of blocks. We computed the average intensity and intensity variance of each block and quantized both to 4 bits. Using a training set of originals and copies we determined the average variance of the blocks at each location. The variances are used as individual block weights, with low variances giving high weights and high variances giving low weights to emphasize the stable blocks. Two images are compared through corresponding inter-block distances. Each block distance is obtained by first summing the L_1 distances between both intensity values and between both variance values, after which the block weight is applied. To distinguish a copy from a non-copy all block distances are added together and thresholded.

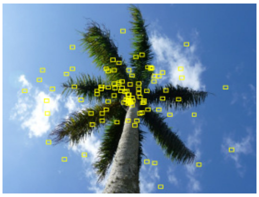


Figure 6.7: The blocks are placed according to an exponential distribution.

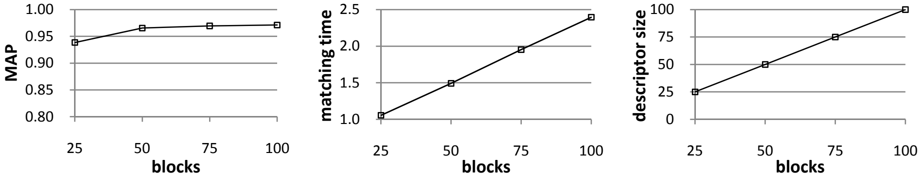


Figure 6.8: Accuracy in mean average precision (left), matching time per query image in seconds (middle) and descriptor size in bytes per image (right) vs. varying numbers of blocks.

6.2.7 Retina-median hybrid

This copy detection algorithm combines the small memory footprint of the median method with the human vision-based approach of the retina method. Using the exponential distribution of the retina method 64 fixed block locations are selected in the image, with blocks being allowed to partially overlap. As a result of preliminary experimentations we found that the best performance was obtained using an image resolution of 640x640 with a block size of 32x32. The image description and copy detection are then carried out in the same way as the median method.

6.3 Image collections

In our experiments, which are discussed in detail in Section 6.4, we will use three image databases: (i) the *query and copy* collection, (ii) the *web* collection and (iii) the *flickr* collection.

6.3.1 Query and copy collection

The query collection consists of 6000 color photos taken at various locations in the world, with sizes alternating between 640x480 and 480x640, depending on whether the photo was taken in landscape or portrait orientation. See Figure 6.9 for example images.

Figure 6.9: Example images from the query collection.

We held a survey to determine what kind of alterations are considered to be duplicates. Because we were interested in not only large but also very small image differences, we contacted people who are familiar with image retrieval, image editing and/or image creation. We asked them to use their favorite search engine to find colored images of any kind and report on the ways duplicates in the search results differed from the query image. Together the 45 respondents looked up 443 images and found 2019 copies. The results are presented in Table 6.1. Our user-oriented approach is similar to the work of Foo et al. [2007a]. Several other studies [Ke et al. 2004; Nikolopoulos et al. 2010] use the 40 transformations proposed by Meng et al. [2003], of which many transformations are similar to ours but some are different, for example rotated and mirrored images were also considered to be copies. Wang et al. [2006] only regard images that are scaled, converted to grayscale or converted into another image format as duplicates of an original image.

Table 6.1: Differences per transformation category between 433 original images and their 2019 duplicates. Note that a duplicate may differ from an original using multiple transformations.

transformation	percentage	totals	transformation	percentage	totals
color to grayscale	1.8	37	flipping	0.4	8
intensity	34.4	694	image format (jpeg, gif, etc.)	1.1	23
hue	2.0	41	framing (black bars)	27.5	555
saturation	1.2	24	rotation	0.3	7
contrast	1.0	21	small logos or text	24.1	486
cropping	31.8	642	large logos or text	13.4	270
despeckling	0.5	10	other	0.8	16
size/resolution	38.5	777			

To create the copies for each of the query images, we focused on those transformations that had an occurrence of at least 1%. For our study we have created 60 copies and can categorize them as follows.

6.3.1.1 Image compression

This category includes changes in the original that result from image compression. These operations change the color values of the pixels, albeit sometimes only modestly, as a result of reducing the information used to reconstruct the color values. For our tests we save the original images at various levels of compression using the Independent JPEG Group scales.

6.3.1.2 Image scaling

For this category we create copies by resizing the original images. In addition we have transformations that squash the image along the horizontal or vertical dimension.

6.3.1.3 Image framing

This category includes various transformations to frame the topic of interest. We applied cropping, where we cut off parts of the image along one or both dimensions. Similarly, letterboxing transformations are also applied by adding black borders around the image.

6.3.1.4 Image colorization

Several transformations are used that adjust the color values of the pixels in the image. One transformation converts the image to grayscale and another reduces the number of colors to 256, similar to a GIF image. Additional transformations adjust the intensity (brightness) of the image or increase the contrast. Because several of the survey respondents remarked that hue and saturation were difficult to tell apart, we only used saturation to adjust the colors of the image in additional transformations.

6.3.1.5 Image elements

We created transformations that overlay logos and/or copyright texts onto the image. Furthermore we have transformations to mimic the use of an image as the background of a web page, overlaying decorative lines or menu items and icons. The exact transformations are listed in Table 6.2 and a selection of them is shown in Figure 6.10. Our query collection is a set known not to be published on the internet, i.e. none of the query images can also be found in the web and/or flickr collections. This ensures that the only duplicates encountered during the experiments will be the images generated by the transformations on the original query image.

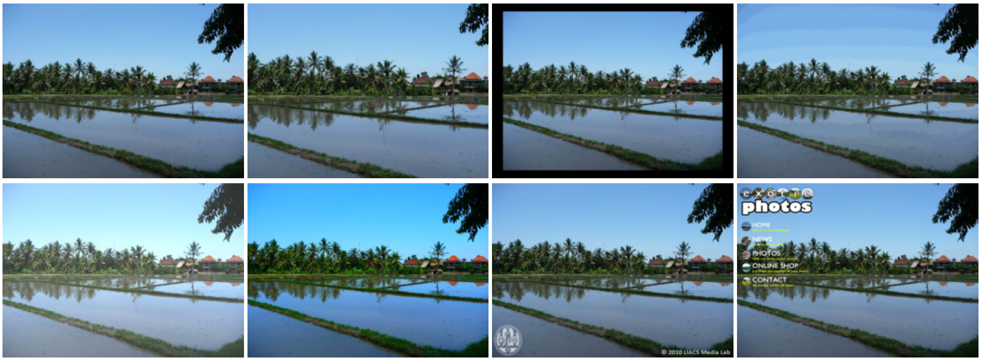


Figure 6.10: Examples of image transformations. Clockwise, starting from the top-left: original image, crop, letterbox, convert to 256 colors, menu, copyright, increase in saturation, increase in intensity.

Table 6.2: Overview of image transformations.

image compression	10 transformations	compress 95% to 50% in steps of 5%
image scaling	13 transformations	scale 20% to 200% in steps of 20%, squash 5% and 10% along width or height
image framing	12 transformations	crop 5% or 10% along width and/or height, add black border 5% or 10% along width and/or height
image colorization	16 transformations	convert to grayscale, reduce colors to 256, brightness +10% to +50% and -10% to -50% in steps of 10%, contrast +10% and +20%, saturate +50% and +100%
image elements	9 transformations	add small or large copyright logo and/or text, add decorative lines, add simple or elaborate menu

6.3.2 Web collection

In total 2,000,000 web images of diverse modality, e.g. logos, graphics, celebrity shots and stock photography, were collected using our internet crawler that we discuss in more detail in Appendix A. In Figure 6.11 we show some example images. The web images represent well what one would generally encounter when browsing on the internet, with the exception of small icons and banners that have been left out. The images have dimensions ranging between 250 and 640 pixels, with an average size of 450x400.

Figure 6.11: Example images from the web collection.

6.3.3 Flickr collection

We downloaded 2,000,000 images from the Flickr website, which is an image and video hosting site where many people share their personal photographs. These images are mostly of highly photographic nature and often very artistic. See Figure 6.12 for several example images. The images have dimensions ranging between 50 and 500 pixels, with an average size of 460x400.

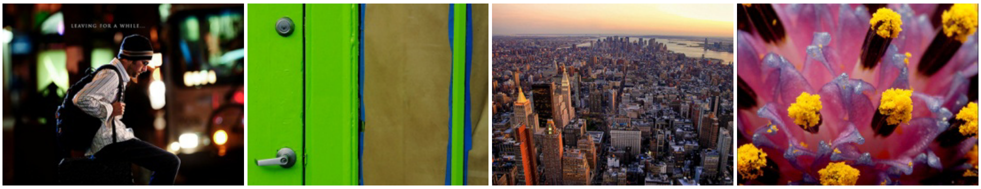


Figure 6.12: Example images from the flickr collection.

6.4 Experiments

In total we used 6000 query images, 216,000 duplicates, 2,000,000 web images and 2,000,000 flickr images. All three image collections were split in half, where one half was used by the methods that required training (i.e. the discrete wavelet transform, color histograms, interest points and retina methods) and the other half was used for testing all methods. In our experiments we compared each query image with either the web database or the flickr database, augmented with its copies. We specifically kept the web and flickr collections separate in order to see which, if any, differences arise when evaluating a method. This is interesting from the point of view that the images in the flickr collection are of highly photographic nature, just like the query collection, whereas the web collection contains images of various modalities.

6.4.1 Computational performance

We focus on three main indicators of performance: *descriptor size* (the amount of memory needed per image), *description time per image* (the average processing time needed to calculate the descriptor of an image) and *matching time per query image* (the average processing time needed to compare one query image with all 1 million web/flickr images plus its copies). Together these measurements constitute the most important method-dependent factors determining requirements for main memory, disk storage and processing times.

The main memory requirements are to an important extent determined by the size of the indexing structure used, if any at all. For indexing many different approaches can be used [Böhm et al. 2001]. In related work we see that hashing-based techniques (e.g. locality sensitive hashing [Yang et al. 2009]) and tree-based techniques (e.g. KD-trees and -forests [Aly et al. 2009]) are popular. The idea of indexing is simple: instead of having to compare a query image with every single image in the database to find the relevant ones (in our case all duplicates), the indexing algorithm performs culling to identify only a fraction of all images which supposedly at least contains all relevant images; indexing thus effectively reduces the time required, since the query image needs to be compared to fewer images, and requires less data to be held in memory, since only the descriptors of the culled set of images have to be loaded in memory instead of those of all database images. Advanced indexing and data structures may improve performance, and we plan to evaluate them in future work.

In Table 6.3 we show the computational performance results for all near-duplicate detection methods. We can see that the median and retina-median methods only require 8 bytes for an image descriptor. In contrast, the wavelet, interest points and color histograms methods need at least 100 times as much memory. As an illustration of memory consumption: for the color histograms method storing one million image descriptors requires 1GB of memory, whereas the median method only needs 8MB. For most methods calculating an image descriptor is quite fast, with the median method being the fastest and the interest points method the slowest. As for matching, the cosine method needs little time to compare image descriptors, whereas the interest points method and particularly the color histograms method are quite slow due to having to calculate the difference between histograms for each pair of images to compare.

The training phase required for the discrete wavelet transform, color histograms, interest points and retina methods resulted in spending additional time, which is shown in Table 6.4. The time required for the training phases of the color histograms and retina methods was relatively short, since both the color histograms' probability mass function and the retina's variances are based on differences between originals and copies. These methods did not need to consider the web and flickr training sets. In contrast, the training phase of both the wavelet and interest points methods took more time, because they required the analysis of all training images to determine the wavelet's shape filter thresholds and the interest points' visual word clusters. The training phase of the interest points method lasted especially long, taking roughly one week on a high-performance blade server and requiring 28GB RAM to calculate the 200,000 visual words.

Table 6.3: Computational requirements per method. Descriptor size is in bytes, description time is the average time in seconds to extract one descriptor, and matching time is the time in seconds to match one query image with all images in the test collection.

method	descriptor size	description time	matching time
discrete cosine transform	35	0.03	0.7
discrete wavelet transform	804	0.07	1.0
interest points (TOP-SURF)	800	0.34	2.2
color histograms	1024	0.04	6.0
median	8	0.02	0.8
retina	50	0.06	1.5
retina-median hybrid	8	0.30	1.0

Table 6.4: Training time in hours.

method	training time
discrete wavelet transform	26
interest points (TOP-SURF)	166
color histograms	1
retina	1

6.4.2 Accuracy

To evaluate the accuracy of the near-duplicate detection methods we use a testing framework that for each query image measures the distances to all images in the test collection. Ideally, all duplicates have small distances to the query image, whereas all other images have large distances. For the success of a method, its accuracy is of paramount importance: if the accuracy is low then the method is useless, even when it demonstrates excellent computational performance. We have measured the accuracy of all methods involving the transformations mentioned in Section 6.3.1 for both the web and flickr collections. For clarity, in our results we define *precision* as the number of copies found over the total number of images looked at and *recall* as the number of copies found thus far over the total number of existing copies. The *mean average precision* (MAP) is often seen as the area under the precision-recall curve and is calculated by first determining the average precision over all copies for each of the queries and then averaging these average precision values.

To analyze the difference in performance on both collections, we calculated the MAP values for each of the methods using all copies, see Table 6.5. As can be seen, the values obtained on the web collection are slightly higher than those obtained on the flickr collection, which suggests that the methods were better able to find the duplicates in the web collection than in the flickr collection. This may be caused by the multiple image modalities present in the web database, making it somewhat easier to discard images as potential duplicates due to their noticeable non-similarity to the query images. Nonetheless, in the remainder of our discussion we will use the average of all results obtained on both collections for each of the methods, since the differences in accuracy between the web and flickr collections are relatively small.

Table 6.5: Mean average precision of each method for the web and flickr collections using all copies.

method	web collection	flickr collection
discrete cosine transform	0.589	0.588
discrete wavelet transform	0.784	0.776
interest points (TOP-SURF)	0.987	0.986
color histograms	0.852	0.847
median	0.885	0.862
retina	0.967	0.964
retina-median hybrid	0.935	0.922

In Figure 6.13 we show the precision-recall curves for each of the transformation categories. In the compression category all methods perform very well, with only the median method performing slightly less and the color histograms method dropping off in accuracy towards the end. In the scaling category all methods also perform well, although the cosine method has difficulty with some of the transformations. The framing category proves rather difficult for most methods, with the cosine method failing badly, although the interest points, retina and particularly the color histograms methods achieve high accuracy. The colorization category affects many methods substantially, with the exception of the median, retina-median and interest points methods. For the color histograms method the color alterations differed too much to match most copies to the trained color distribution. In the elements category the cosine method once again does not have high accuracy and the wavelet method has trouble with two of the transformations. The other methods perform reasonably well, with the retina, interest points and color histograms methods coming out on top. As can be noticed, some of the curves are not very smooth, e.g. the retina method in the colorization graph, which is because the various transformations in a particular category do not necessarily range from 'easy to detect' to 'difficult to detect'. Some duplicates may be considered to be equally hard to detect from the point of view of one method, whereas this is different for another method. Overall, if we look at all categories combined, we see that most methods perform quite well, with the interest points method performing best and the cosine method performing worst. The interest points method performed so well because its descriptor is robust to significant image content alterations.

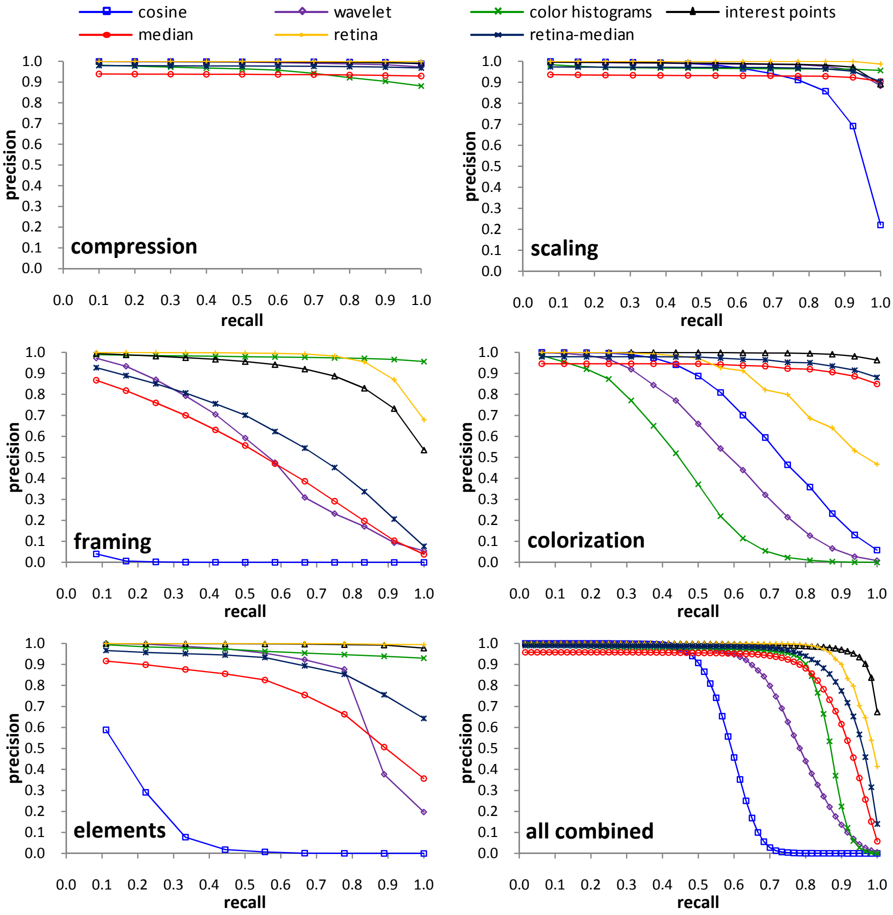


Figure 6.13: Precision-recall curves of all methods for each of the transformation categories: compression (top-left), scaling (top-right), framing (middle-left), colorization (middle-right), elements (bottom-left), all categories combined (bottom-right).

6.4.3 Ranking

To see on which transformations the methods perform particularly well or bad, we show the average rank at which a duplicate is detected in Figure 6.14. Ideally the rank at which a duplicate is found is 1. In the compression category we see that as the amount of compression increases (copy 1 is slightly compressed, copy 10 is highly compressed) that the rank generally also increases. When we look at copies of different sizes (copies 11-19) and those that are squashed (copies 20-23), almost all methods perform fairly constant, with an exception for the cosine method that improves as the images get larger. Interestingly, the cosine method prefers images being squashed in height (copies 21 and 23) rather than in width (copies 20 and 22). For most methods the average rank increases when images are increasingly cropped (copies 24-29) and when the border gets thicker (copies 30-35). The cosine method is simply not suited for this category.

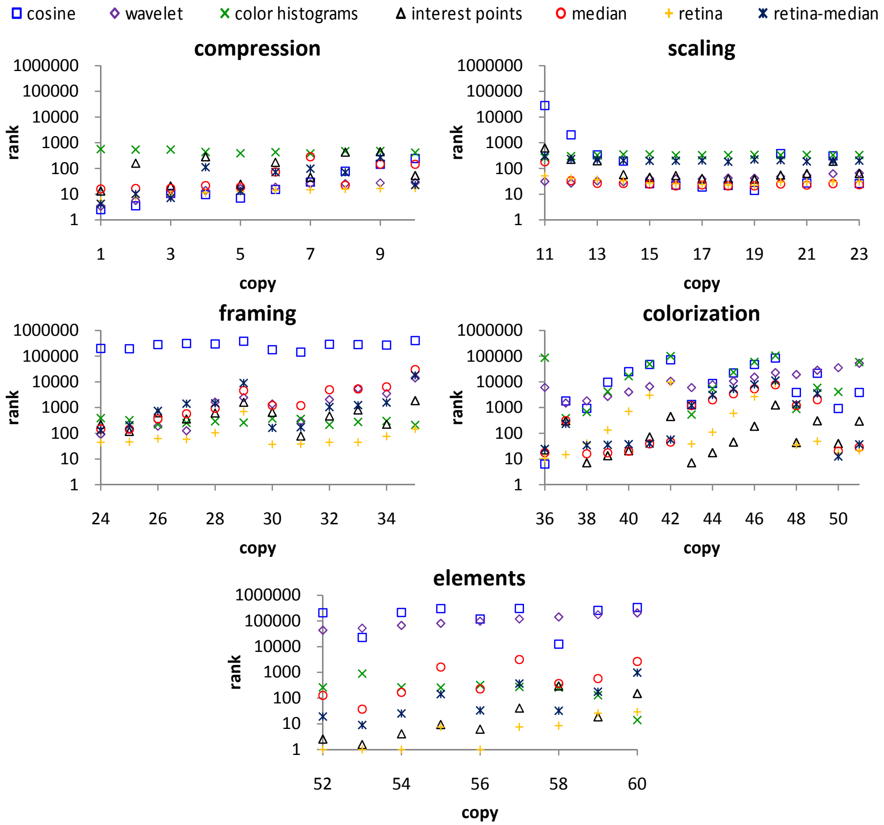


Figure 6.14: Average rank per copy for each of the transformation categories: compression (top-left), scaling (top-right), framing (middle-left), colorization (middle-right), elements (bottom). The numbers along the copy axis refer to the transformations in the order as specified in Table 6.2.

In the colorization category we see that most methods can handle the conversion to grayscale well (copy 36), with the exception of the wavelet and color histograms method, since these are heavily dependent on color. The conversion to 256 colors (copy 37) has a large negative effect on most methods, with the exception of the retina method. As the brightness ranges from brighter to very bright (copies 38-42) and darker to very dark (copies 43-47) we see the all methods perform increasingly badly. The median, retina-median and interest points methods are least affected by an increase in brightness, whereas the retina method and particularly the interest points method are least affected by an increase in darkness. Most methods have trouble with an increase in contrast (copies 48 and 49), except for the retina and interest points methods. The retina, median, retina-median and interest points methods have the least problems with an increase in saturation (copies 50 and 51). Finally, in the elements category we see that the interest points method and especially the retina method have less trouble with the addition of a small copyright logo/text (copies 52-54) and are not much affected by a larger logo/text (copies 55-57). The cosine and wavelet methods perform badly in this category, also for the decorative lines (copy 58) and the menu overlays (copies 59 and 60).

6.4.4 Discussion

An ideal near-duplicate image detection method needs little time to calculate the descriptor of an image and this descriptor uses a minimal number of bytes. In addition, the time needed to compare this descriptor to other image descriptors is also short. Finally, the accuracy of the method must be very high. However in practice no method possesses all these properties. It is therefore important to realize the tradeoffs between the various performance indicators and weigh them accordingly to the intended application needs. Since we are using large image databases, the following are the points of attention for us: (i) if the descriptor size is large, then the method will cause memory consumption issues, unless certain measures are taken (e.g. indexing), (ii) if the description time is long, then the time necessary to calculate all descriptors will become prohibitive, (iii) if the matching time is long, then real-time requirements will suffer (e.g. performing on-the-spot detection of copyright infringement for a given original image), and, most important of all, (iv) if the accuracy is low then duplicates either won't be found or non-duplicates will be incorrectly labeled as duplicates. In Figure 6.15 we combine Table 6.3 and Table 6.5 in graphical form. In each of these graphs the ideal descriptor would be located in the top-left corner.

If we evaluate the results with these tradeoffs in mind we can deduce the following. First, the discrete cosine transform-based method only works well for small changes in image content. However, for a range of transformations that are commonly found on the internet this method is not suitable. Second, the color histograms method roughly detects each duplicate around the same rank, but this rank is simply too high resulting in many non-copies being marked as more copy-like than true copies. Its high memory requirement does not lead to high accuracy, and its matching time is too long for the method to be useful. Similarly, the discrete wavelet transform also uses a large image descriptor, but its accuracy is generally average and it only performs well on the compression and scaling categories. The median method shows quite good overall accuracy, especially considering that it only requires 8 bytes for its image descriptors. For a moderate increase in descriptor size and matching time the retina method performs better. As a blend between the retina and the median methods, the retina-median method disappoints somewhat. Rather than improving upon its parent methods by borrowing their good aspects and leaving out the bad ones, its accuracy ends up between both of them. Yet, this method's performance is generally good, but its long description and matching times make it not very attractive. Finally, if accuracy is the most important factor and good computational performance plays a smaller role then the interest points method is the best choice, since it outshines the competition by a large margin.

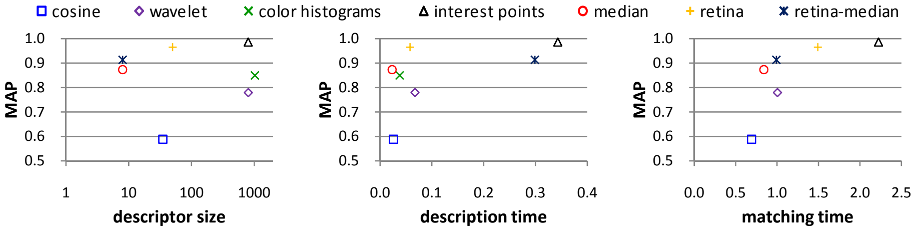


Figure 6.15: Performance tradeoffs using accuracy in mean average precision vs. descriptor size in bytes (left), vs. description time of an image in seconds (middle) and vs. matching time per query image in seconds (right). Note that the color histograms' matching time falls off the chart.

To put the performance of near-duplicate detection methods into perspective we can look at ourselves. It is clear that the human vision system is the benchmark for visual matching, yet unlike computers none of us will tirelessly browse through millions of images to find copies of a particular image. Even then, the time needed to do so will be orders of magnitude longer than the slowest of the methods we evaluated. The human vision system is very complex and we may be never able to fully replicate its functioning into a set of algorithms. Nonetheless, it serves as a fantastic inspiration to visual retrieval techniques, as we have seen in this paper with the high performance of the retina and interest point methods. This notwithstanding, the average rank at which a copy is detected using the best near-duplicate detection methods is often not the best rank possible. Many non-copies are therefore incorrectly labeled as copies. When looking at millions of images this number of false-positives can be quite substantial, e.g. reaching a precision of 99.9% on one million images still means that on average there are one thousand false positives.

6.5 Conclusions

In this chapter we have compared several content-based near-duplicate image detection methods and assessed their performance in the context of internet search on representative databases in total containing over 2 million images. We have shown that to obtain high accuracy it is not necessary to use a large nor computationally intensive image descriptor. We also presented results per transformation category to gain further insight into the strengths and weaknesses of the candidate methods. Based on the obtained results we can conclude the following: (i) if very low memory usage (8 bytes per image) and fast matching are of paramount importance, while still obtaining good accuracy, then the median method is the best choice, (ii) if low memory usage (50 bytes per image) and high accuracy are the most important, then the retina method is preferred, and (iii) if medium memory usage (800 bytes per image) is acceptable and accuracy is the most important factor, then the interest points (TOP-SURF) method is the best technique to use.

A. Noteworthy image search

We have developed an image retrieval system using many of the techniques we have proposed in this thesis. The search engine aims to provide a personalized search experience and focuses on returning images that are noteworthy (interesting, special) to the user. One of the goals of the search engine is to find the newest images for any particular query, for example *"what are the newest images of 'James Bond' since my last visit?"*. Because search results often contain many near-duplicate images, we integrated a copy detection technique in order to diversify the image results. We discuss in detail how the search engine was designed and developed, and offer insight into techniques for handling large amounts of images.

A.1 Introduction

With current estimates putting the number of images available on the internet into the tens of billions, it is a immensely interesting resource. These images are very diverse and include tiny icons, advertisements, celebrity photos, travel photos and even NASA satellite imagery. There are many search engines available that allow people to find such images, ranging from well-known ones like Google Image Search and Picsearch to the lesser-known ones like Pixsy. Almost all search engines support keyword-based searches to find images, where images are associated with descriptive keywords that are usually extracted from the webpages they were found on. Not many commercial search engines support content-based searches, although Google Images is currently rolling out a feature to enable searching for images that look similar to one of the images shown on screen. Another search engine called Riya, which was recently discontinued, used face recognition software to detect which people were present in photos uploaded by users, and could also learn to recognize new faces.

We noticed that current internet search engines do not offer duplicate detection and advanced personalization of the search results. With the creation of our image retrieval system we aim to fill this void. In the following sections we will introduce our search engine and discuss in detail how it was designed and developed. First of all, in Section A.2 we present our internet crawler for indexing and downloading the millions of images available on the internet and offer insight into techniques for handling large amounts of images. In Section A.3 we describe our approach for allowing the user to search for images by keyword and in Section A.4 show how the search engine handles similar and duplicate imagery. Because the search engine is still under development, we have not completed our goal to offer the user a personalized searching experience. In Section A.5 we therefore present our personalization plans for the future, especially on how to return 'noteworthy' imagery to the user. All techniques come together in the search engine we describe in Section A.6. While our internet crawler navigated the internet we gathered interesting statistics, which we present in Section A.7. Finally, we conclude in Section A.8.

A.2 Internet crawler

One of our first goals was to download as many images as we could find on the internet, and do this as fast as possible. The fact that the internet is very large and constantly changing makes this a formidable task. With only limited quantities of storage, memory and processing power available, it requires a prudent approach to make this task manageable. We discovered from our early prototypes that using ‘modern’ techniques, such as threads, fibers and in-memory communication, resulted in unstable behavior, where often a single thread that misbehaved (e.g. crashed, used the CPU or memory excessively) could bring down the entire internet crawler. Therefore we shifted our focus to proven techniques and our current internet crawler uses separate processes. The processes only communicate with each other using files on disk and do not interfere with each other in any other way. Since direct communication is not required this is a solution that is both elegant and robust. In Figure A.1 we show an overview of our internet crawler.

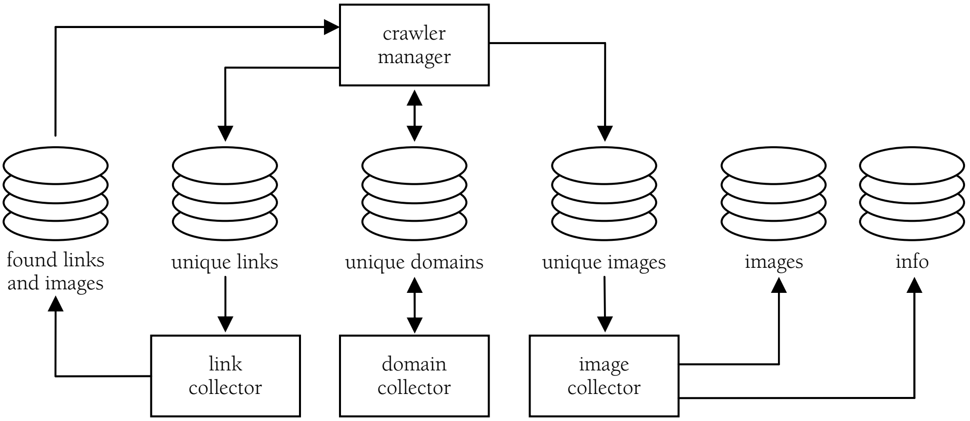


Figure A.1: The internet crawler.

One of the main tasks of the crawler manager is to prepare files containing unique items for the collectors to look at. The domain collector requests each domain’s robots.txt file, which is a file that indicates which parts of the domain may be looked at by an internet crawler and which parts may not. If a domain is off-limits to our crawler, it will be placed on the black list, so that any future URLs that point to that domain can be ignored. The domain collector is also responsible for tracking how often requests to a particular domain succeed or fail, and if the success-failure ratio is low the domain will also be moved to the black list. The link collector analyzes webpages and extracts all URLs that point to other links and to images. In the case of an image URL, it additionally obtains raw information that describes the title of the webpage, the text surrounding the image, and so on. The idea is that all this information might refer to the actual content of the image. The image URL and the image information are passed to the image collector, which downloads the image. To ensure the images are likely to be of interest to the user, any small icons or banners are discarded, and all other images, together with their raw information, are stored on disk.

The manager performs a uniqueness check of all URLs discovered by the link collector, so that no URL is visited twice. Storing each complete URL in memory and comparing them character by character is infeasible storage- and processing-wise. Therefore we represent each URL by hashing it to a 64-bit signature, using an adapted version of the Message-Digest algorithm 5 (MD5), a widely used, but partially insecure, cryptographic hashing function. Its strength is that highly similar inputs will hash to completely different outputs, and additionally it will rarely occur for URL signatures to collide. Binary trees are very suitable for quickly looking up whether or not a signature exists. However, during preliminary experimentation we discovered that using a single binary tree to store all link and image signatures caused the lookups to dramatically slow down over time as the number of entries greatly increased. Assigning one binary tree per domain to store the signatures in resulted in faster lookup speed, although the trees still grew to a large size for popular domains. At the same time we noticed that the content at these popular domains is frequently updated. As a joint solution to both observations, our final solution was to use two binary trees per domain, one for the link signatures and one for the image signatures, and to remove all entries from the link tree once a day. By frequently emptying the link tree we retain fast link signature matching, while simultaneously ensuring we can revisit all webpages sufficiently often to check for new imagery. By not emptying the image tree, thus keeping all image signatures, we still prevent the same images from being downloaded more than once. Because URLs found on a page often point back to another page on the same domain, the manager randomizes the order in which they are distributed to the link and image collectors, to avoid accessing the same web server too many times in succession.

A.3 Keywords

The raw image information needs to be processed before it can be used by the search engine. Without also analyzing the image content it is not possible to say with certainty to what extent the information correctly describes what is shown in the image. However, there are certain indicators that tell which pieces of information are likely to be more correct than others. For instance, it is more probable that the filename of the image refers to what the image is about than a random word taken from the text surrounding the image. To distinguish between these types of information, we therefore assign each of them a *confidence factor*. These factors will be used when producing the image ranking to emphasize those images with a higher confidence of matching the query keywords. This is discussed in more detail in Section A.6. The keywords are stored on disk using an inverted index, i.e. each keyword file contains the indices of the images that it is associated with, including their confidence factors, making it easy to look up which images are associated with the query. An overview of the keyword extractor is shown in Figure A.2.

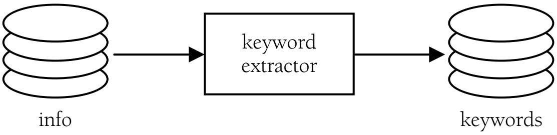


Figure A.2: The keyword extractor.

To support searching by quote, our keyword extractor combines one or more successive keywords, so that users can, for instance, search for images that are associated with the phrase *"star wars"*, rather than for images that are only related to the individual words *'star'* and/or *'wars'*. With the latter query there is thus no guarantee that both words were associated with the image and, even if that is the case, they may not have been found on the webpage in the specified order. We also support the common additive and subtractive operators, so that users can force the inclusion or absence of certain keywords, e.g. *"star wars" +yoda –"luke skywalker"* ensures that any returned images are associated with the phrase "star wars" and the word 'yoda', but not with the phrase "luke skywalker". The keyword extractor additionally filters out very common words and very short words, because these are unlikely to result in matches that the user is particularly interested in. As a special functionality, we have built in preliminary support for language detection, so that the keyword extraction can be done more efficiently and people will be able to use the language of their choice when searching. Overall, these customizable search options give the user a lot of control and using such special queries is likely to improve the quality of the search results as experienced by the user.

A.4 Similarity and duplicates

In our research on near-duplicate image detection (Chapter 6) we demonstrated how suitable our TOP-SURF descriptor (Chapter 5) is for finding copies and near-copies of images. Because the search results of popular image search engines often include many of such images, there is not much diversity in the images shown. For instance, when querying Google Image Search for the painting *"mona lisa"* we were shown 20 images, of which 8 were virtually the same and the other 12 were very similar to each other. It would be arguably better if the user is also shown other results, such as images of the Louvre (where the painting is on display), or of Leonardo da Vinci (the painter). Once such diverse results are shown, if the user is really only interested in a certain image she can request the system to show her with more similar ones. To give our system this functionality we have integrated a similarity and duplicate detection technique, of which an overview is shown in Figure A.3.

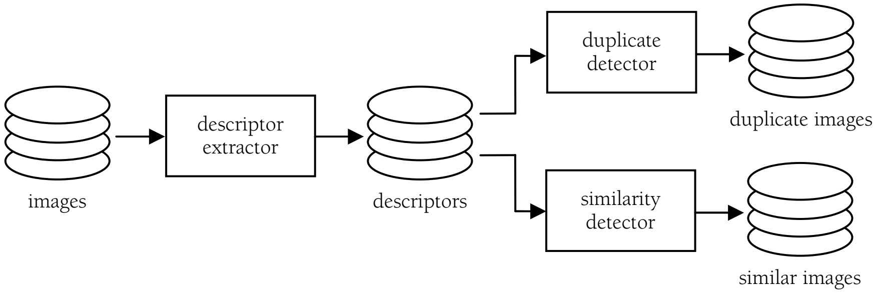


Figure A.3: The descriptor extractor and the duplicate and similarity detectors.

The TOP-SURF descriptors are extracted and stored in a database, after which they are analyzed by both the duplicate detector and the similarity detector. The difference between both detectors is that the duplicate detector enforces a strict threshold that ensures very high similarity between images before considering them as copies, whereas the similarity detector does not perform any thresholding, allowing for the creation of a ranking of images that ranges from highly similar to hardly similar.

A.5 Personalization and noteworthiness

To serve the user better, our search engine ideally gets to know each individual user, so that it will always return those images that the user is interested in. We introduce the notion of *noteworthiness* of an image, which refers to the intrinsic qualities of an image that somehow make it special from *the perception of the user*. To take it to the extreme, for a movie fanatic user anything related to movies is noteworthy and everything else is not, whereas for a sports fan all sports-related imagery will be noteworthy. For the general public, however, determining the noteworthiness of an image will not be so black and white. As we mentioned in the introduction, our search engine is still under development and the personalization aspect is not yet completed. We believe analyzing user search behavior is the key to answering the question of what makes an image noteworthy and will investigate this in the near future. Since noteworthiness is subjective, it will most likely be defined as a combination of what the population in general finds noteworthy and what a certain person finds noteworthy. One direction we may look into is using neural networks to learn to discover and detect when an image is noteworthy or not.

Another direction we will look at is how to adjust the search results based on past search behavior. An example would be that if the user has previously searched for technology and now wants to find images related to *"snow leopard"*, our search engine would display results for the Apple operating system before any other results. If, on the other hand, the user has issued animal- and nature-related queries in the past, images of the real snow leopard would be shown first. Currently we already allow users to register with our search engine for improved search functionality. An interesting feature offered by the search engine is to keep a timeline of the queries a user has issued, so that it can give her the newest images for any particular query since her previous visit. We intend to expand the number of personalization features in the future to further enhance the search experience.

A.6 Search engine

In the search engine all mentioned techniques come together, see Figure A.4. When the user types in a search query, the search engine looks up all images that are associated with each of the keywords in the query and present the results in a ranking. Each image receives an overall score that is based on the number of keywords (or phrases) that an image matches and their confidence scores. The higher the score, the higher the image will be in the ranking. At the same time, the search engine looks up all copies of the images that are about to be shown and if necessary modifies the ranking when duplicate images are discovered. Once the first page of results is presented the user has several options, i.e. the user can (i) click on an image to visit the page on which it was found, (ii) view all images that are similar to an image, (iii) view all copies of an image, and (iv) ask the search engine to return another page of image results.

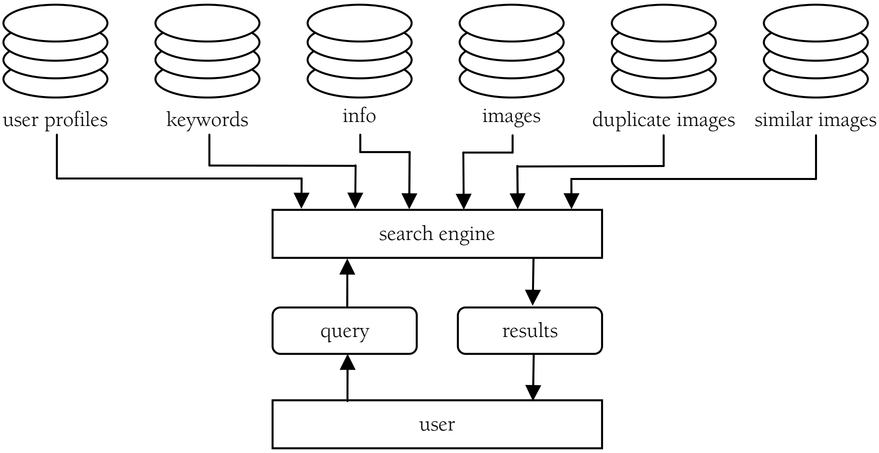
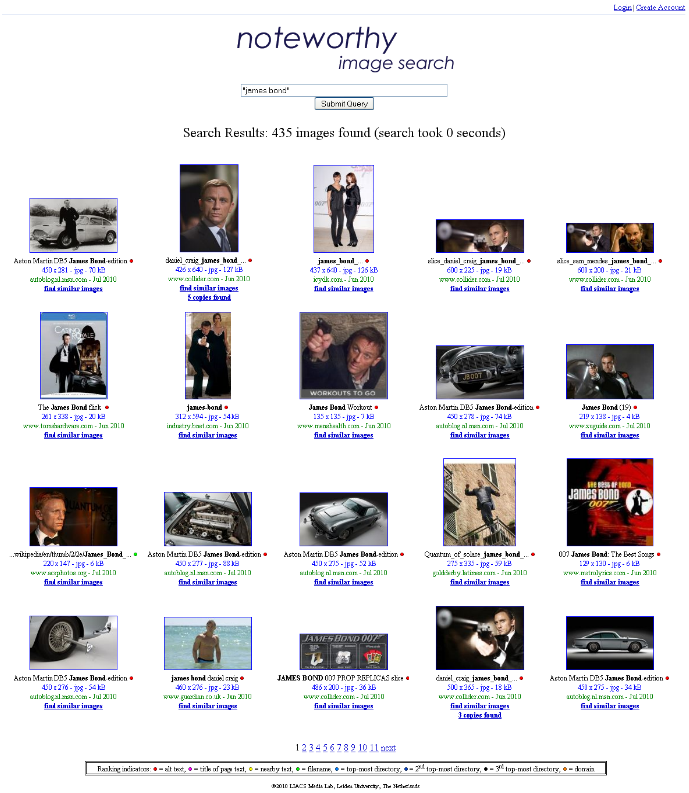


Figure A.4: The search engine.

Similar to existing search engines we look up a piece of descriptive text for each returned image and highlight any query terms appearing in that text. Our search engine looks through all types of information associated with the image (e.g. title, text surrounding the image) and selects the one that contains one or more query terms at the highest confidence. By doing so the user obtains a basic understanding why a particular image was returned by the search engine. As a novel feature we have improved the clarification of why an image was selected through the use of a notification system based on colored labels. Each piece of descriptive text is accompanied by a label that visually indicates which type of information is responsible for matching the user's query. When the mouse cursor hovers over the label, an additional textual clarification is presented. The textual clarifications are also shown at the bottom of the page. As far as we know such a clarification system is the first of its kind. The system informs the interested user in more detail why a particular image was chosen, this in contrast with existing search engines, where it is often not clear what the main motivation of the search engine is for returning an image in response to a query.



Search Results: 435 images found (search took 0 seconds)

Aston Martin DB5 **James Bond**-edition ●
450 x 281 - jpg - 70 kB
autoblog.nl.msn.com - Jul 2010
find similar images

daniel_craig_**james_bond**_... ●
426 x 640 - jpg - 127 kB
www.collider.com - Jun 2010
find similar images
5 copies found

james_bond_... ●
437 x 640 - jpg - 126 kB
icydk.com - Jun 2010
find similar images

slice_daniel_craig_**james_bond**_... ●
600 x 225 - jpg - 19 kB
www.collider.com - Jul 2010
find similar images

slice_sam_mendes_**james_bond**_... ●
600 x 200 - jpg - 21 kB
www.collider.com - Jun 2010
find similar images

The **James Bond** flick ●
261 x 338 - jpg - 20 kB
www.tomshardware.com - Jun 2010
find similar images

james-bond ●
312 x 594 - jpg - 54 kB
industry.bnet.com - Jun 2010
find similar images

James Bond Workout ●
135 x 135 - jpg - 7 kB
www.menshealth.com - Jun 2010
find similar images

Aston Martin DB5 **James Bond**-edition ●
450 x 278 - jpg - 74 kB
autoblog.nl.msn.com - Jul 2010
find similar images

James Bond (19) ●
219 x 138 - jpg - 4 kB
www.zuguide.com - Jul 2010
find similar images

...wikipedia/en/thumb/2/2e/**James_Bond**_... ●
220 x 147 - jpg - 6 kB
www.acephotos.org - Jul 2010
find similar images

Aston Martin DB5 **James Bond**-edition ●
450 x 277 - jpg - 88 kB
autoblog.nl.msn.com - Jul 2010
find similar images

Aston Martin DB5 **James Bond**-edition ●
450 x 275 - jpg - 52 kB
autoblog.nl.msn.com - Jul 2010
find similar images

Quantum_of_solace_**james_bond**_... ●
275 x 335 - jpg - 59 kB
goldderby.latimes.com - Jun 2010
find similar images

007 **James Bond**: The Best Songs ●
129 x 130 - jpg - 6 kB
www.metrolyrics.com - Jun 2010
find similar images

Aston Martin DB5 **James Bond**-edition ●
450 x 276 - jpg - 54 kB
autoblog.nl.msn.com - Jul 2010
find similar images

james bond daniel craig ●
460 x 276 - jpg - 23 kB
www.guardian.co.uk - Jun 2010
find similar images

JAMES BOND 007 PROP REPLICAS slice ●
486 x 200 - jpg - 36 kB
www.collider.com - Jul 2010
find similar images

daniel_craig_**james_bond**_... ●
500 x 365 - jpg - 18 kB
www.collider.com - Jun 2010
find similar images
3 copies found

Aston Martin DB5 **James Bond**-edition ●
450 x 275 - jpg - 34 kB
autoblog.nl.msn.com - Jul 2010
find similar images

1 2 3 4 5 6 7 8 9 10 11 next

Ranking indicators: ● = alt text, ● = title of page text, ● = nearby text, ● = filename, ● = top-most directory, ● = 2nd top-most directory, ● = 3rd top-most directory, ● = domain

©2010 LIACS Media Lab, Leiden University, The Netherlands

Figure A.5: The search interface showing the results of the query "james bond".

Our interface is shown in Figure A.5 and is intentionally designed to look like the interface of Google Image Search to provide the user with a sense of familiarity. In our interface, the user can type in keywords in the query box to perform a search. Once images are returned by the search engine, the user can click on one of them to navigate to the original URL at which it was found, find similar images or inspect any copies that might have been found. When the shown images are not satisfactory, the user can also browse to view more image search results using the numbered pane at the bottom of the interface. The user can opt-in to be recognized by the search engine, so that her previous queries are remembered and the newest images that have appeared on the internet since the last visit can be shown. When the future personalization ideas we discussed in the previous section have been fully developed and integrated into the search engine, real personal-based search will become available to the users.

A.7 Statistics

While crawling the internet we gathered many interesting statistics. In Figure A.6 we show the number of successful and unsuccessful connections made for the 10,000 most popular domains. The most popular domains were those that had the highest number of URLs pointing at them. We excluded any domain that had a negative connection rate, i.e. more unsuccessful than successful connections. The 10 most popular domains are listed in Table A.1.

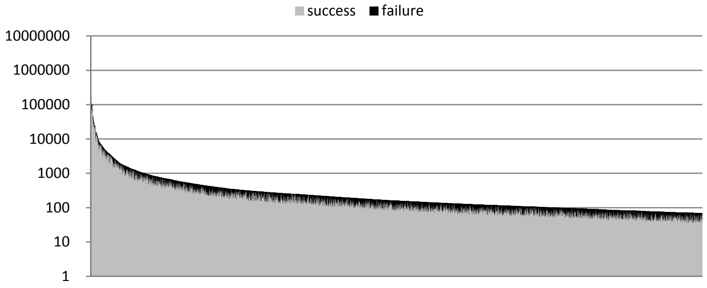


Figure A.6: Number of successful and unsuccessful connections for the top 10,000 domains.

We investigated the causes for connections to be unsuccessful and show the errors that occurred in Figure A.7. As can be seen, the most common reason a connection was unsuccessful was because the URL to visit belonged to a domain that was blacklisted and the connection was therefore prevented from being made. Other errors we see are network connectivity errors, such as not being able to contact the server, and content errors, such as images not being large enough.

Table A.1: The 10 most popular domains, including the number of successful and unsuccessful connections made.

#	domain	successful	unsuccessful
1	amazon.com	984137	42353
2	tomshardware.com	389217	12394
3	twitter.com	339092	18830
4	news.com	294044	10739
5	informationweek.com	228138	25335
6	billboard.com	176433	14585
7	tvguide.com	176841	2168
8	washingtonpost.com	166057	11603
9	metrolyrics.com	139106	3295
10	youtube.com	116436	1757

One of the most important aspects of an internet crawler is the number of URLs discovered and how fast it can visit them. In Figure A.8 we show the number of links, images and domains our crawlers have discovered, how many they have visited and the average speed of doing so. We can clearly see that the beginning was very tumultuous, with very high speeds being reached for the number of links discovered, up to 500 links/sec. At that moment, the available network bandwidth was the bottleneck. Shortly thereafter the number of new links, images and domains discovered decreased rapidly, and this time the crawler manager was the bottleneck. Note that the first big drop in speed shown in the graphs occurred, because we temporarily turned the crawler off to check its logs. The general decline in speed is due to the manager performing uniqueness checks of all URLs in receives, and all collectors find themselves waiting for the manager to assign them URLs to visit. We can also see in the figures see that the number of links discovered is many times larger than the number of links the link collectors are able to visit. In contrast, the image collectors are able to visit all discovered images and the domain collectors are able to visit all discovered domains.

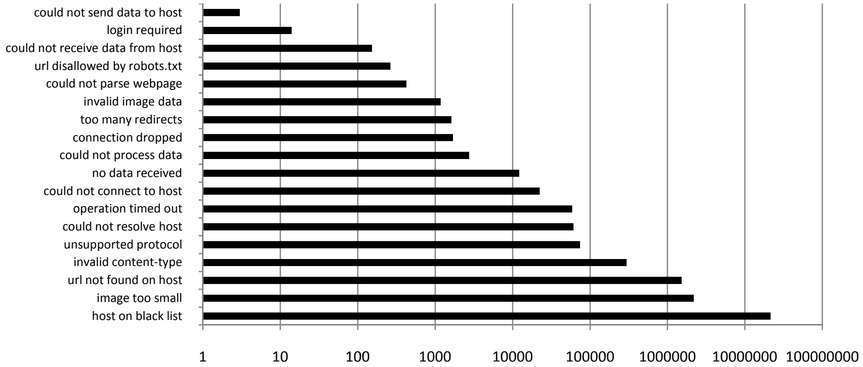


Figure A.7: Number of unsuccessful connection attempts specified per type of error.

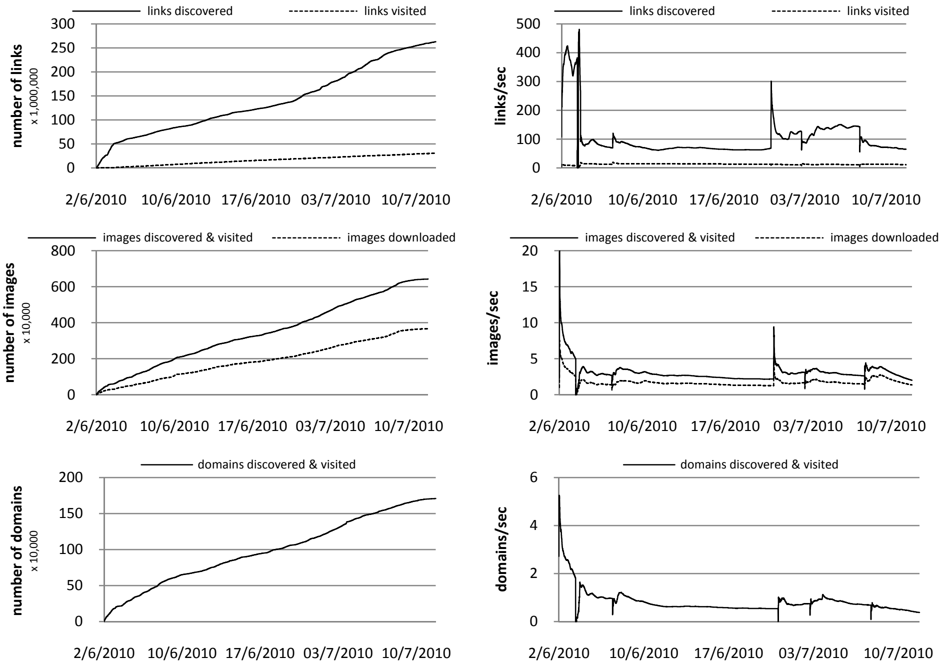


Figure A.8: Number (top-left) and speed (top-right) of links discovered and visited, number (middle-left) and speed (middle-right) of images discovered, visited and downloaded, number (bottom-left) and speed (bottom-right) of domains discovered and visited.

We can observe that around 3 July a significant event took place, because it caused a boost in the collection speed. That day we incorporated a new policy to empty the link binary tree on a daily basis. This resulted in the link collectors visiting many URLs that had been discovered in the past and now were allowed to be accessed again. We can see that over time these speed boosts became less pronounced. This is because the emptying of the binary tree is handled individually per domain and takes place when the domain is accessed for the first time after the expiry date. The moment of accessing can be at any moment in time, thus such speed boosts do not occur simultaneously anymore. Nonetheless, the speed of crawling generally is declining. On the last day in the figures, 13 July, on average only 65 links were discovered and 11 were visited per second. On the same day a mere average of 2.0 images were discovered, 1.4 images were downloaded and 0.4 new domains were visited per second. Future work should address this issue, so that speeds will be significantly higher and ideally remain constant.

A.8 Conclusions

In this appendix we presented an image search engine that is still under development. One of its most important building blocks are the images that can be searched for. At this moment the performance of the internet crawler that collects these images is not yet satisfactory and requires improvement. The near-duplicate detection technique we integrated into the search engine is a novel addition that provides diversity in the search results, because duplicate images can be grouped together. Another novel feature is our label-based clarification system that allows the user to get a better understanding of why images were returned by the search engine given a particular query. In the near future we will further look into personalization of the search results and what makes images noteworthy.

B. Touch-up! image search

Taking the artificial imagination approach we proposed in Chapter 3 a step further, we present our early work on an interactive image retrieval system that is based on the principles of scene completion. The search engine allows the user to erase unwanted parts of an image and have them replaced by content that she is interested in. The idea is that these realistically-looking touched up images will lead to better retrieval results, because they more closely match what the user is looking for.

B.1 Introduction

The initial query image is of paramount importance when searching for images of interest, because it has a direct effect on the retrieval results. The image that is chosen as the query image is of course similar to what the user is looking for, but nonetheless often does not fully represent what the user has in mind. The selected image for instance may contain elements that are not relevant to the user's intention. Our assumption is that the presence of these elements negatively impacts the performance of the search engine, causing the retrieval results not to be as good as the user would like them to be. One reasonable explanation is that this happens because the user looks at an image from a high-level, i.e. the query image contains the general *concept* the user is interested in, even though this is not necessarily completely reflected by its *content*. For instance, suppose that the user is looking for images that show a view of the Grand Canyon and she uses the image shown in Figure B.1a. To us it is obvious her intention is that the car should not be considered by the search engine, rather only the scenery from the other parts of the image. Yet, traditional search engines will have no idea of the user's particular intention and simply use all image content. In the most positive scenario the retrieved images may be other scenes containing a similar view with a car, although it is certainly possible that they might as well be scenes that focus on cars, or be scenery that contains a large green object. Either way, the presence of the car is very likely to have a negative effect on the retrieval results from the point of view of the user. Our aim is thus to allow the user to erase such unwanted elements (Figure B.1b), and to let the resulting hole be seamlessly filled with semantically valid (i.e. plausible, realistic) content that more closely reflects what she has in mind (Figure B.1c). In a sense the retrieval system is thus artificially imagining scenes. In comparison with the images that are returned by the search engine when using the original (unmodified) scene as the query image, we expect that the search engine will return more relevant images when the hole-filled scene is used instead.

Because our scene completion algorithm is still under development, our touched up scenes currently do not yet look plausible enough. We aim to have completed the search engine in the near future, so we can perform experimentation to test our hypothesis. In this appendix we present the current state of our work and offer insight into the techniques used by our retrieval system.

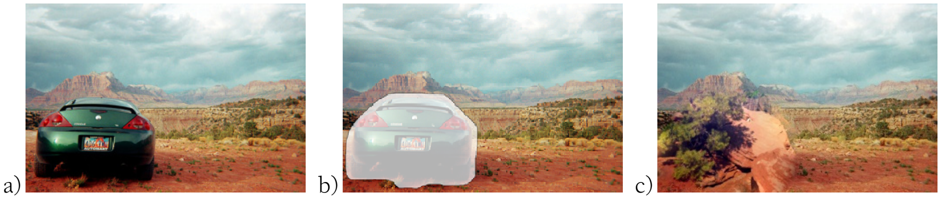


Figure B.1: Erasing unwanted elements from photos: a) original scene, b) scene with the car and its shadow masked out, c) touched up scene where the car is seamlessly replaced by a rock and a tree.

B.2 Scene completion

To allow the user to erase undesired parts of a query image and to consequently fill the resulting missing pixels with desired content, we incorporate the scene completion technique that was proposed by Hays and Efros [2007] into a search engine. In contrast with other scene completion techniques [Criminisi et al. 2003; Wilczkowiak et al. 2005], their method gives overall more plausible results. In human user experiments, they showed that the percentage of users thinking their completed scenes were fake was significantly lower (~62%) compared to percentage of users thinking the same of scenes completed using the algorithm of Criminisi et al. (~90%). Yet the percentage is still much higher than that of real photographs considered to be fake (~12%). The scene completion technique attempts to find semantically similar scenes to the incomplete query image and identifies the best image patch that matches the context around the missing region. This patch is then seamlessly blended into the image to complete the scene. In the following sections we will discuss the technique in more detail.

B.2.1 Image collection

Hays and Efros discovered that using a small image collection of 10,000 images for finding suitable image patches led to unsatisfactory results, due to the most similar scenes often not being similar enough. They noticed that as the number of images in the collection increased, the quality of the completed scenes increased as well. The most likely explanation for the improved quality was that with more images to choose from it is more likely to find a very similar scene to the query image. The authors finally gathered a collection of 2.3 million images. To cover for a wide range of suitable hole filling material, the collection consisted of travel-related imagery (e.g. landscapes and cityscapes) downloaded from Flickr.

B.2.2 Finding suitable image patches

For an image patch to be blended into the incomplete scene in a plausible way, it must fulfill three conditions. The image patch must be:

1. *graphically* valid, i.e. the texture gradient at the edges of the patch must closely correspond with the gradient around the missing region in order to seamlessly blend the two together,
2. *contextually* valid, i.e. it must be taken from an image that is similar in appearance to the image that is to be completed,
3. *semantically* valid, i.e. its content must make sense when viewing the completed image as a whole.

These three conditions each operate on a different level of image analysis, with the graphical condition operating on the low-level, the contextual condition on the mid-level and the semantic condition on the high-level. When the conditions are not met, the completed scene is affected as follows: (i) when the graphical condition is violated the image patch will not blend in smoothly and hard edges will be noticeable, (ii) when the contextual condition is violated the content in the image patch will be discolored, as a result of the blending process, and (iii) when the semantic condition is violated the content of image patch will look unnatural in combination with the content of the incomplete scene. See Figure B.2 for illustrations of the condition violations.

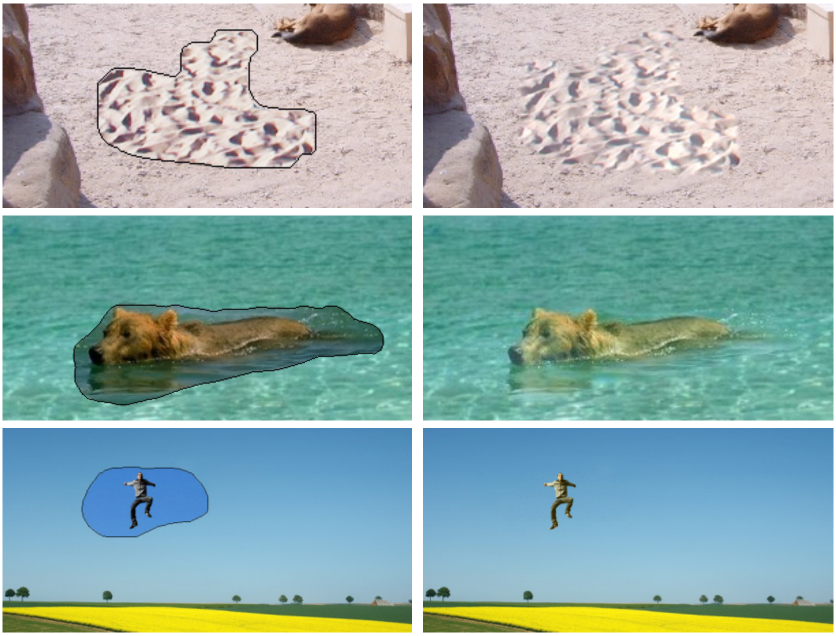


Figure B.2: Image patches that violate the graphical condition (top, the beach sand is too rough compared with the sand in the query image), contextual condition (middle, the brown bear turns greenish) and semantic condition (bottom, a cloud in the blue sky is replaced by a jumping person) when blended into an incomplete scene.

B.2.2.1 Finding similar scenes

To find suitable image patches the idea is to first find scenes from the image collection that are contextually and semantically similar to the incomplete image. Ensuring graphical validity will take place during the local context matching, which is described in the next section. To determine the most similar scenes the gist scene descriptor [Oliva and Torralba 2001, 2006] is used in combination with a descriptor in L*a*b* ('lab') color space. The gist descriptor aggregates oriented edge responses at multiple scales into very coarse spatial bins. Hays and Efros found that a gist descriptor built from 6 oriented edge responses at 5 scales aggregated to a 4x4 spatial resolution to be most effective. The lab descriptor contains the color information of the scene resampled to the spatial resolution of the gist. Both descriptors complement each other very well, since the gist descriptor is suitable for finding similarly *structured* scenes, whereas the lab descriptor is suitable for finding similarly *colored* scenes. In Figure B.3 both descriptors are visualized. Although both are low-level descriptors and inherently cannot describe higher level contextual and semantic information, the assumption is that, because of the large number of images in the collection, the best matching scenes will be very similar in structure and color to the incomplete scene and thus will fulfill the first two required conditions.

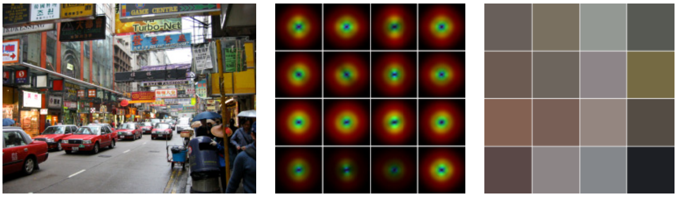


Figure B.3: An example image (left) and its gist (middle) and lab (right) descriptors.

When the user provides a query image with a missing region, first its gist and lab descriptors are computed, after which its similarity to each of the scenes in the image collection is determined. The ideal matching scene is the one of which the context corresponds with the incomplete query image, i.e. both (i) the structure of the two images and (ii) the coloring of both images should be similar to each other, *excluding* the hole. The reason for excluding the hole is that in principle both the structure and the color of the masked out query patch do not have to match those of the candidate image patch, because any valid content can be placed there. To achieve this, a separate weight mask is applied that weights each spatial bin in proportion to how many valid pixels are in that bin. Since the spatial resolution for both the gist and lab descriptors is identical, the same weight mask can be used.

Because the distance range of both descriptors is different, a training set of images is used to normalize the distances by weighting them so that their standard deviations are equal and they thus influence the ranking similarly. In the final distance calculations the gist distance contributes twice as much as the lab distance. Of all the millions of images in the collection, only the 200 best matching ones are used for fine-grained matching.

B.2.2.2 Local context matching

Because the gist descriptor ensures that the best matching candidate scenes are similar in structure to the query image, and the lab descriptor ensures that they are similar in coloring, the next step is to perform fine-grained matching. This is done by aligning the candidate scenes to the so-called *local context* around the masked out region. The local context is defined as an 80 pixel wide boundary on the outside of the mask, see Figure B.4. Because the candidate scenes can be of a different size (i.e. resolution-wise) and scale (i.e. zoomed in/out) in comparison with the query image, the local context is floated over the scenes at three different scales and by translating it in all directions. The contexts are pixel-wise compared in L*a*b* color space and also using their coarse texture gradients. Because the gist descriptor already ensured the scenes and the query were already roughly aligned, distant matches receive a penalty based on their displacement offset.

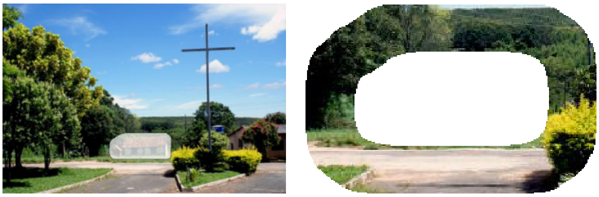


Figure B.4: An input image with a masked out region (left) and its local context (right).

B.2.3 Seamless blending

Simply copying an image patch into the missing region will not give a plausible result. Even though the alignment of the local context means that the structure and coloring near the boundaries of both the patch and the hole are roughly the same, the transition between the two is not smooth enough and will be noticed by the user. A graph cut and image blending technique is used to solve this issue.

B.2.3.1 Graph cut seam finding

When the optimal image patch has been found for each of the candidate scenes, the graph cut seam finding technique of Kwatra et al. [2003] is applied to find the best seam that reduces hard edges and artifacts as much as possible. This is achieved by minimizing the gradient of image differences along the seam using the graph cut algorithm of Boykov et al. [2001]. The graph cut algorithm is allowed to cut from both the query image and the image patch, although cutting from the query image is discouraged by using a penalty for removing each pixel that increases with the distance from the hole. Pixels can only be cut along the seam, whereas on the outside of the seam the pixels must come from the query image and on the inside of the seam the pixels must come from the image patch. An illustration is shown in Figure B.5.

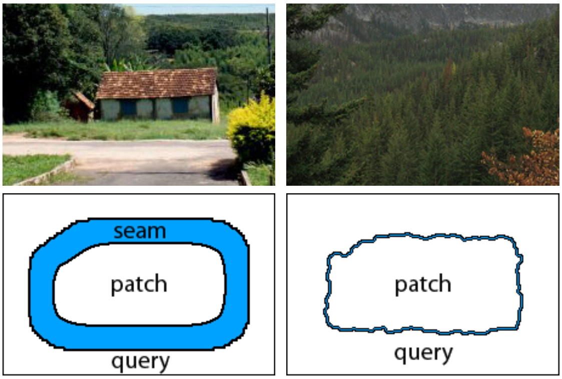


Figure B.5: Graph cut seam finding, showing a close-up of the query image around the missing region (top-left), the candidate image patch (top-right), the graph cut labeling indicating that cutting is only allowed along the seam, and the resulting optimal seam (bottom-right).

B.2.3.2 Poisson blending

Given the optimal seam, the image patch still cannot be copied directly into the query image. Even though the gradients along the seam have been minimized, they are not yet nicely aligned. The seamless cloning technique of Pérez et al. [2003] is used to smoothly blend the incomplete scene and the image patch together into a composite scene. This is done by performing a guided interpolation of the gradient of the image patch using the gradient of the missing region as the guidance field, i.e. the gradient of the image patch is made to look like the gradient of the query patch that it is replacing. The interpolations to be solved are Poisson equations with Dirichlet boundary conditions, which are to be uniquely and independently solved in each of the RGB color channels. Pérez et al. applied either Gauss-Seidel iteration with successive over-relaxation or V-cycle multigrid to solve the equations, whereas Hays and Efros used the Poisson solver of Agrawal et al. [2006]. An illustration of Poisson blending is shown in Figure B.6.

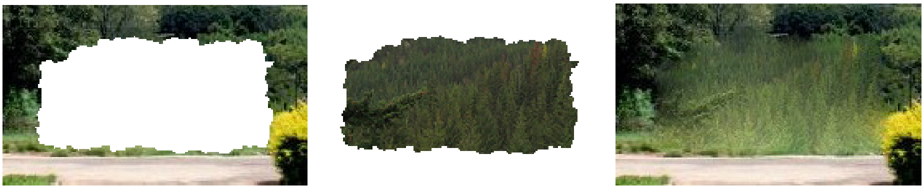


Figure B.6: Blending the query image region (left) and image patch (middle) to obtain the composite patch (right). As can be seen, the colors of the image patch are adjusted by the blending process to match those used in the query image, resulting in a smooth and hardly noticeable seam.

The final completed scenes, one composite scene per query image/image patch combination, are ranked based on the combined score obtained by adding the scene matching distance, the local context distance and the cost of the graph cut, normalized so that each component influences the score equally. The 20 scenes with the lowest scores are then shown to the user.

B.2.4 Algorithm optimization and modification

In our work, we have implemented the scene completion technique to the best of our ability, based on the sequence of steps and values used as described in the original paper. Our end goal was to increase the computational performance of the algorithm to near real-time using a single high-end workstation, so that a user does not have to wait more than a few seconds before the completed scenes are displayed. Hays and Efros used a cluster of 15 machines to perform the scene completion, with matching and blending of the scenes taking several minutes. We optimized and slightly modified the algorithms to speed up the scene completion and to further improve the quality of the resulting scenes. The most notable changes we made are the following:

Image collection
We enlarged the collection even further to 3 million images. Having access to more images should increase the likelihood of finding even better matching image patches. In contrast with the images downloaded by Hays and Efros, which had a maximum dimension of 1024 pixels on either side, we downloaded images with a dimension of 500 pixels, because these were more readily available.

Descriptor normalization
Hays and Efros used a set of training images to normalize the descriptor distances by weighting them so that their standard deviations are equal. Our normalization approach is similar, albeit slightly different. We randomly selected 10,000 images from our collection of 3 million images and for each descriptor determined the average distance μ and the standard deviation σ resulting from comparing all images with each other. We then considered each descriptor's distance weight ratio dwr as

$$dwr = \mu + 1.5\sigma \ . \tag{B.1}$$

The normalization factor nf, with which the regular descriptor distance will be multiplied, is its inverse

$$nf = \frac{1}{dwr} \ . \tag{B.2}$$

The normalization factors ensure that the adjusted descriptor ranges account for most of the variability in the data. After applying the normalization factors the descriptors are compatible with each other, e.g. when the gist and lab distances are combined, a 10% change in the gist descriptor has about the same impact as a 10% change in the lab descriptor.

The gist descriptor used for selecting the candidate scenes has a fixed length and therefore its normalization factor only needs to be calculated once. Even though the lab descriptor has a fixed length for selecting the candidate scenes, its length varies when it is used for matching the local context. The number of pixels in the local context – and thus the length of the descriptor – is dependent on the mask the user has drawn. As a consequence its normalization factor is also variable. This is also the case for the texture descriptor. We therefore determined the relation between the distance weight ratios of the lab and texture descriptors and the number of pixels in an image. We extracted the descriptors from images of varying sizes, namely 32x32, 64x64, 96x96 and 128x128 pixels. Note that since the L*, a* and b* color components of the lab descriptor have different value ranges, each component also has a different impact on the distance when comparing two descriptors. We therefore calculated the normalization factors for each of its color components first, before determining the overall normalization factor for the lab descriptor. The experimental results, shown in Figure B.7, revealed a linear relation for both the lab and texture descriptors, allowing us to easily obtain suitable normalization factors for local contexts of any size.

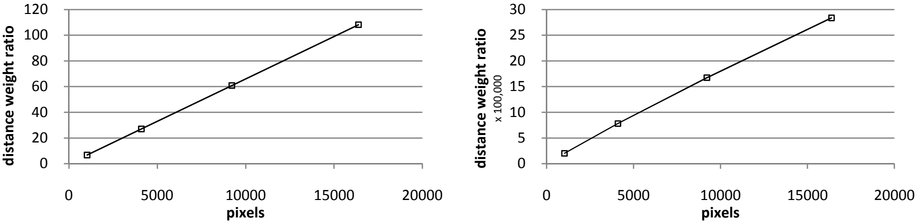


Figure B.7: Distance weight ratio for various numbers of pixels for the lab descriptor (left) and the texture descriptor (right).

Image matching and indexing

Because the scenes only partially use the gist and descriptors, depending on which regions of a query image are missing, it is not straightforward to apply an indexing technique to speed up the discovery of suitable candidate scenes. However we may be able to loosen this restriction a bit, particularly regarding the lab descriptor. The Poisson blending algorithm adjusts the colors of the candidate image patch to more closely resemble those used in the query patch that it is replacing. When their coloring differs greatly, the colors in the image patch may be shifted unnaturally, see for example Figure B.8. Therefore it may be wise not to apply the weighted mask to the lab descriptor when matching candidate scenes, forcing candidate scenes to have a similar colored matching region to the query region. Of course, this may not always be the desired behavior, for instance in Figure B.1 it is not desirable to replace the green car by something else that is green, but rather something that matches the orangeness of the Grand Canyon. However, due to the characteristics of the Poisson blending process, any image patch to replace the car will take on a greenish tinge. Alternative blending approaches may be the solution to solve this discoloration issue, for instance by applying the seamless image compositing algorithm of Guo and Sim [2009], an example of which is shown in Figure B.9. We will look into such alternative blending techniques in the future.

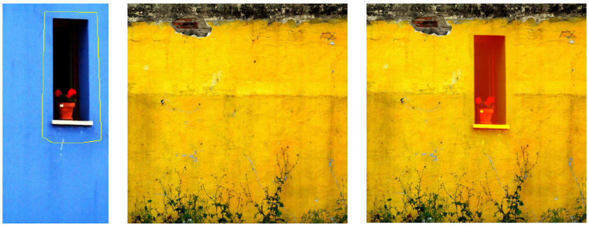


Figure B.8: Poisson blending can cause the coloring of the window (left), when it is placed on the yellow wall (middle), to shift unnaturally (right).

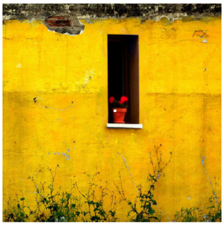


Figure B.9: The seamless image compositing algorithm of Guo and Sim [2009] retains the coloring of the window when it is placed on the yellow wall.

If we use the lab descriptor in its entirety, it is possible to apply indexing on the descriptor. We therefore implemented an approximate nearest neighbors technique that is based on a forest of randomized kd-trees [Muja and Lowe 2009]. This technique is the same as the one we used in Chapter 5. From the 3 million images we then first obtain the 10,000 approximate nearest neighbors according to their lab descriptor, after which we more accurately match each of them using both the gist and lab descriptors, finally narrowing the number of candidates down to the best 200. In contrast with linear search, this approximate matching technique should significantly speed up the candidate scene matching.

Local context

Because the lab and gist descriptors use very coarse bins, and because the scenes may differ in resolution, it is unclear how to correctly determine the initial location in a candidate scene from where to start matching the local context. In our approach we used the relative location of the center of the local context in the query image to determine its corresponding location in the candidate scene, e.g. if the original local context is positioned at 75% of the width and 60% of the height of the query image, then this will correspond with a location in the candidate scene at 75% of its height and 60% of its width. In contrast with the original local context matching, where all valid translations were considered for which the local context was fully contained within the candidate scene, we restrict the amount of translation in all directions. This is because (i) the gist descriptor already ensured that the local contexts are already roughly aligned, so distant matches are unlikely to be ever chosen, and (ii) comparing the contexts for all valid translations within all candidate scenes is computationally very intensive. As a consequence of this rough estimation where to place the local context in the candidate scene, we decided not to use the translation offset as a penalty score and simply focused on selecting the best matching local context. Note that we reduced the size of the local context boundary from 80 to 40 pixels, since the images we downloaded are smaller than those Hays and Efros used.

Selecting the final scenes
In the original algorithm, the distances of the candidate matching, the local context matching and the cost of the graph cut are all combined, all with an equal importance on the overall score. Normalizing the descriptor distances was relatively straightforward, however this is not the case for the graph cut, because it not only depends on the number of pixels in the local context, but also the number of pixels on the edge of the context, the intensity differences between the pixels to cut, and so on, making it difficult to determine what the relation is between the normalization factor and the shape of the graph. Rather than tackling this problem directly, we devised an alternative approach to allow for an equal importance for all three scoring components. We simply assign a rank to each completed scene per component and reach its combined score by adding the three independent ranks. For instance, if a particular scene has the fifth lowest distance for the candidate matching, the second lowest distance for the local context matching and the third lowest graph cut cost, its final score will be $5 + 2 + 3 = 10$. The 20 completed scenes with the lowest overall scores are then returned to the user.

At the moment our scene completion algorithm is nearing completion, and the quality of the initial completed scenes is promising, although the scenes are not yet sufficiently plausible. Hays and Efros noted that their algorithm performed poorly when only having access to a limited number of candidate scenes. For this reason we downloaded 3 million travel-related images from Flickr. Upon closer inspection of our image collection we noticed that a very large percentage of these images were in fact not the type of images we expected to obtain, but rather were incorrectly tagged. This may be one of the causes why our completed scenes are not realistic. As a consequence, we made changes to our Flickr crawler to ensure it only downloads images of the appropriate kind and recently restarted it. In the near future we hope to have again collected several million images, now true travel-related images, which we hope will lead to better completed scenes. As we mentioned before, we have adjusted the original algorithm on several accounts, mainly focusing on making the algorithm faster. We already noticed that our algorithm only takes a few minutes to complete 20 scenes on a single workstation, in contrast with the original algorithm that, according to Hays and Efros, took one hour to complete. We will investigate in the future the extent of the impact that all our changes have had on the quality of the completed scenes.

B.3 Searching for images

The scene completion technique has so far only been used to generate scenes where the missing region has been filled in with semantically meaningful content. Our idea is to take this a step further and integrate the technique into the process of image search to improve the retrieval performance. We consider the technique to be particularly interesting, because it allows the user to remove unwanted elements from the query image *before* submitting it to the retrieval system. Our hypothesis is that by removing these elements the search engine will be able to return better images than it would have done when leaving the image unchanged. Of course, even though the image that is chosen by the user as the query image is already similar to what the user is looking for, it may be that any non-relevant image elements are still taken into account by the retrieval system, negatively affecting the retrieval performance. In the next sections we will outline our approach for such a scene completion-based image retrieval.

B.3.1 Keyword-based search

Common ways for a retrieval session to start is with the user providing an example image [Bian and Tao 2010] or by typing in one or more keywords [Lu et al. 2003]. It is also possible that the user chooses one from a random selection of images from the database [Thomee et al. 2009b]. Having to provide an example image is often very inconvenient for the user, because this means that the user should already have an image available that is similar to the one she wants to find. Choosing an image from a random selection is more convenient, but may require requesting the system to show additional sets of random images before an image similar to the target image is located. In contrast, the option to search using keywords is a very attractive way to start a new retrieval session, since the user simply has to type in one or more words that describe the image of interest.

With keyword-based retrieval it is very important that images are associated with appropriate words that describe them, because images will not show up if they are mislabeled or not labeled at all. Many image collections are already annotated or make it possible to obtain the annotations if necessary. For instance in a collection of images found on social photo sharing websites, such as Flickr, keywords can be trivially obtained, since users typically assign a number of descriptive tags to the images they upload to the site. To enable a keyword-based search on unannotated collections, it is generally not feasible to manually annotate all images as collections often contain thousands or millions of items. Therefore more and more research is directed at automatic image annotation [Fan et al. 2008]. Although the performance of the current techniques is not completely satisfactory yet, the results are promising and the quality of the annotations is likely to greatly improve over the next years.

For our retrieval system we designed an internet crawler that visited as many websites as possible and downloaded all images it encountered, under the condition that they were big enough (e.g. no icons) and regularly shaped enough (e.g. no banners). To obtain descriptive keywords to enable searching by text we analyzed the webpage on which each image was found. We specifically focused on the title of the page, the filename of the image, the text surrounding the image and so on, because all this information might refer to the actual content of the image. We assigned a confidence value to each extracted keyword to indicate its likelihood of actually referring to the image content. The confidence values were based on where on the webpage the keyword was found, since it is for instance more likely that the filename of the image correctly describes the content of the image than one of the words in the surrounding text. Our image crawler is discussed in more detail in Appendix A. In total we downloaded 10 million images.

When using the search engine the user can enter one or more words and/or phrases. The system will then look up all images that are associated with them and present the results in the form of a ranking. The more keywords an image matches, and the higher their associated confidence values, the higher in the ranking the image will be.

B.3.2 Content-based search

We consider that searching by keyword is a good way to start the search, but not for finding all images that the user is looking for. As mentioned before, images that are precisely what the user is looking for will not be found via a text-based search if they are not properly annotated. It is a well-known fact that human annotation is subjective, thus in collections that are manually annotated the same image may be differently labeled when multiple labelers are involved. We also noticed that user tags on social photo sharing sites are quite frequently incorrect, sometimes caused by batch tagging, i.e. tagging many photos at once without considering each photo individually. Relying exclusively on text is thus not a good strategy, especially when the user is interested in visual components of the image that are difficult to describe with words. The saying *'a picture is worth a thousand words'* is quite appropriate in this context, because it implicitly suggests that a handful of keywords can impossibly accurately and completely describe the content of an image. Therefore we integrate visual matching into our retrieval system, where the content of images is analyzed to determine which ones are similar to the query image and which ones are not. In principle any descriptor that is appropriate for determining visual similarity can be used by the retrieval system. In our retrieval system we may choose between two kinds of descriptors, the first one is the gist+lab descriptor combination and the second one is our TOP-SURF descriptor.

Gist+lab
The gist+lab descriptor combination is a logical choice to use in our search engine, since we have already used them and we know that they are very suitable for finding images that are similar in structure and coloring as the query image. We thus believe these descriptors will allow the retrieval system to return all images of interest to the user. Similar to how they were used in the scene completion algorithm, when the query image is compared with images in the web collection, we determine the similarity between their descriptors using the sum of squared differences. In the final distance calculations we also let the gist distance contribute twice as much as the lab distance. Because there is no hole anymore in the query image there is no need for any local context matching to take place. Finally, the web images are ranked based on their distances to the query image, where those that show the most similarities to the query image are shown at the top of the ranking.

TOP-SURF
We have developed a descriptor called TOP-SURF that is based on SURF [Bay et al. 2008], but is a factor 50 smaller in size. Our descriptor is discussed in more detail in Chapter 5. In essence, TOP-SURF pre-clusters the SURF interest points extracted from a large number of training images into a large number of visual words. Given a new image, the detected interest points are then converted into a frequency histogram of occurring visual words. By applying tf-idf weighting [Salton and McGill 1983] the top N most descriptive visual words are then selected from the frequency histogram as the descriptor of the image. In our experiments on near-duplicate detection, discussed in Chapter 6, we observed that the TOP-SURF performed very well. It will be interesting to see how well it performs to find similar images, rather than copies.

We plan to evaluate in the future how well these descriptors will be able to find similar images to a query image. The best performing one, either gist+lab or TOP-SURF, will be selected for inclusion in the retrieval system.

B.3.3 Touch-up! search

At any time in the search process the user can decide to touch up one of the shown images. The user can mask out parts of the image and let the scene completion algorithm generate a set of plausible scenes. From these generated scenes the user can then choose the composite scene that best corresponds with what she has in mind. In principle, the image can be touched up any number of times until the user is satisfied with the final composite image. This image is then used as the query for the content-based search. An overview of the retrieval system is shown in Figure B.10.

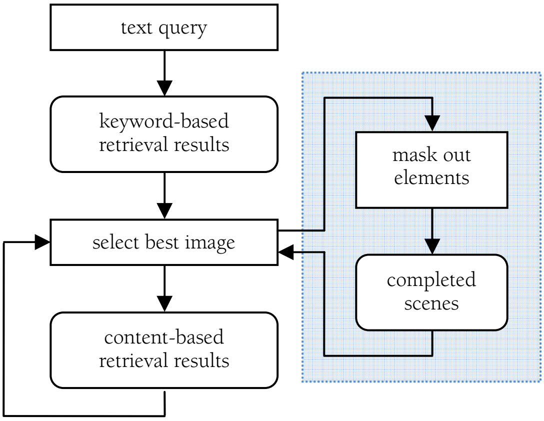

text query
keyword-based retrieval results
select best image
content-based retrieval results
mask out elements
completed scenes

Figure B.10: Overview of our retrieval system, with the scene completion component highlighted on the right hand side.

Note that the image collection in which the user wants to find images is not necessarily the same database that is used for completing the scenes. It is thus possible to perform the keyword- and content-based searches in any database, for instance one of the many versions of the well-known Corel collection (e.g. [Duygulu et al. 2002]), one of the MIR-FLICKR sets (e.g. [Huiskes et al. 2010]) or a proprietary image collection. The main purpose of the travel-related scene completion image database is to fill up any holes the user has created after removing undesired image elements. However, both collections should contain images that are not too structurally different, since the gist descriptor attempts to find the best structurally matching scenes to the query image. If no suitable structural matches can be found the resulting completed scenes are unlikely to look plausible. One important issue we thus have to investigate is whether or not the 10 million web images we downloaded are appropriate for scene completion. Suppose a user finds an image that she likes, but the image requires some touching up to remove an unwanted element. The structure of the image then needs to correspond with several of the scenes in the collection of landscape images in order for the scene completion algorithm to be able to fill the masked region with meaningful content. Our preliminary experiments show that many of the web images are actually quite different from the travel-related scenery for the scene completion algorithm to find good candidate scenes. We therefore may need to focus on obtaining a more similar image collection, thus perhaps solely focusing on images from Flickr instead.

B.4 Conclusions

The scene completion algorithm of Hays and Efros is a promising technique for removing unwanted elements from an image by replacing it with plausible content. We have performed an initial implementation of this technique and integrated it into a keyword- and content-based image retrieval system. Even though our completed scenes look promising, they do not look realistic yet. We are in the process of improving the algorithm, and we expect our algorithm to benefit from the new collection of travel-related images we are currently downloading. Once our algorithm gives satisfactory results, we will perform experiments on a large-scale database to test whether or not touched up images will return more relevant images to the user.

Bibliography

AGGARWAL, G., ASHWIN, T.V., AND GHOSAL, S. 2002. An image retrieval system with automatic query modification. *IEEE Transactions on Multimedia*, 4(2), 201-214.

AGRAWAL, A., RASKAR, R., AND CHELLAPPA, R. 2006. What is the range of surface reconstructions from a gradient field? In *Proceedings of the 9th European Conference on Computer Vision* (Graz, Austria), 578-591.

ALY, M., WELINDER, P., MUNICH, M., AND PERONA, P. 2009. Scaling Object Recognition: Benchmark of Current State of the Art Techniques. In *Proceedings of the 1st IEEE Workshop on Emergent Issues in Large Amounts of Visual Data* (Kyoto, Japan), 1-9.

AMIN, T., ZEYTINOGLU, M., AND GUAN, L. 2007. Application of Laplacian mixture model to image and video retrieval. *IEEE Transactions on Multimedia*, 9(7), 1416-1429.

AMORES, J., SEBE, N., REDEVA, P., GEVERS, T., AND SMEULDERS, A. 2004. Boosting contextual information in content-based image retrieval. In *Proceedings of the 6th ACM International Workshop on Multimedia Information Retrieval* (New York, NY, USA), 31-38.

ASHWIN, T., GUPTA, R., AND GHOSAL, S. 2002. Leveraging non-relevant images to enhance image retrieval performance. In *Proceedings of the 10th ACM International Conference on Multimedia* (Juan-les-Pins, France), 331-334.

BAO, L., CAO, J., XIA, T., ZHANG, Y.-D., AND LI, J. 2009. Locally non-negative linear structure learning for interactive image retrieval. In *Proceedings of the 17th ACM International Conference on Multimedia* (Beijing, China), 557-560.

BAR-HILLEL, A., HERTZ, T., SHENTAL, N., AND WEINSHALL, D. 2005. Learning a Mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6, MIT Press, 937-965.

BARAS, D., AND MEIR, R. 2007. Reinforcement learning, spike time dependent plasticity, and the BCM rule. *Neural Computation*, 19(8), 2245-2279.

BARRETT, S., CHANG, R., AND QI, X. 2009. A fuzzy combined learning approach to content-based image retrieval. In *Proceedings of the 2009 IEEE International Conference on Multimedia and Expo* (New York, NY, USA), 838-841.

BARTOLINI, I. 2006. Context-based image similarity queries. In *Proceedings of the 3rd International Workshop on Adaptive Multimedia Retrieval: User, Context, and Feedback* (Glasgow, UK), 225-235.

BAY, H., ESS, A., TUYTELAARS, T., AND VAN GOOL, L. 2008. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3), 346-359.

BELHUMEUR, P.N., HESPANHA, J.P., AND KRIEGMAN, D.J. 1997. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), 711-720.

BIAN, W., AND TAO, D. 2010. Biased discriminant Euclidean embedding for content-based image retrieval. *IEEE Transactions on Image Processing*, 19(2), 545-554.

BÖHM, C., BERCHTOLD, S., AND KEIM, D.A. 2001. Searching in high-dimensional spaces: index structures for improving the performance of multimedia databases. *ACM Computing Surveys*, 33(3), 322-373.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *Transactions on Pattern Analysis and Machine Intelligence*, 23(11), 1222-1239.

BRADSKI, G.R. 2000. The OpenCV library. *Dr. Dobbs Journal*, 25(11), 120-126.

BRINKE, W. TEN, SQUIRE, D.MCG., AND BIGELOW, J. 2004. Similarity: measurement, ordering and betweenness. In *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems* (Wellington, New Zealand), 169-184.

BRINKER, K. 2003. Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th International Conference on Machine Learning* (Washington, DC, USA), 59-66.

BRODATZ, P. 1966. Textures: a photographic album for artists and designers. Dover Publications.

CAI, D., HE, X., AND HAN, J. 2007a. Spectral regression: a unified subspace learning framework for content-based image retrieval. In *Proceedings of the 15th ACM International Conference on Multimedia* (Augsburg, Germany), 403-412.

CAI, D., HE, X., AND HAN, J. 2007b. Regularized regression on image manifold for retrieval. In *Proceedings of the 9th ACM International Workshop on Multimedia Information Retrieval* (Augsburg, Germany), 11-20.

CAMPBELL, I. 2000. Interactive evaluation of the ostensive model using a new test collection of images with multiple relevance assessments. *Journal of Information Retrieval*, 2, 87-114.

CHA, G.-H. 2003. Bitmap indexing method for complex similarity queries with relevance feedback. In *Proceedings of the 1st ACM International Workshop on Multimedia Databases* (New Orleans, LA, USA), 55-62.

CHAN, C.-H., AND KING, I. 2004. Using biased support vector machine to improve retrieval result in image retrieval with self-organizing map. In *Proceedings of the 11th International Conference on Neural Information Processing* (Calcutta, India), 714-719.

CHANG, E.Y., WANG, J.Z., LI, C., AND WIEDERHOLD, G. 1998. RIME: A replicated image detector for the world-wide web. In *Proceedings of the 1998 SPIE Symposium of Voice, Video and Data Communications* (Boston, MA, USA), 68-77.

CHANG, E.Y., AND LAI, W.-C. 2004. Active learning and its scalability for image retrieval. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo* (Taipei, Taiwan), 1, 73-76.

CHANG, H., AND YEUNG, D.-Y. 2007. Locally smooth metric learning with application to image retrieval. In *Proceedings of the 11th IEEE International Conference on Computer Vision* (Rio de Janeiro, Brazil), 1-7.

CHATZIS, S., DOULAMIS, A., AND VARVARIGOU, T. 2007. A content-based image retrieval scheme allowing for robust automatic personalization. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval* (Amsterdam, Netherlands), 1-8.

CHEN, Y., AND WANG, J.Z. 2002. A region-based fuzzy feature matching approach to content-based image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9), 1252-1267.

CHEN, Y.-S., AND SHAHABI, C. 2003. Yoda, an adaptive soft classification model: content-based similarity queries and beyond. *ACM Multimedia Systems*, 8(6), 523-535.

CHEN, X., ZHANG, C., CHEN, S.-C., AND CHEN, M. 2005. A latent semantic indexing based method for solving multiple instance learning problem in region-based image retrieval. In *Proceedings of the 7th IEEE International Symposium on Multimedia* (Irvine, CA, USA), 37-45.

CHEN, Y., BI, J., AND WANG, J.Z. 2006. Miles: multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12), 1-17.

CHEN, Y., REGE, M., DONG, M., AND FOTOUHI, F. 2007. Deriving semantics for image clustering from accumulated user feedbacks. In *Proceedings of the 15th ACM International Conference on Multimedia* (Augsburg, Germany), 313-316.

CHEN, Y., DONG, M., AND WANG, W. 2009. Image co-clustering with multimodality features and user feedbacks. In *Proceedings of the 17th ACM International Conference on Multimedia* (Beijing, China), 689-692.

CHENG, E., JING, F., LI, M., MA, W.-Y., AND JIN, H. 2006a. Using implicit relevance feedback to advance web image search. In *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo* (Toronto, ON, Canada), 1773-1776.

CHENG, E., JING, F., ZHANG, L., AND JIN, H. 2006b. Scalable relevance feedback using click-through data for web image retrieval. In *Proceedings of the 14th ACM International Conference on Multimedia* (Santa Barbara, CA, USA), 173-176.

CHENG, J., AND WANG, K. 2006c. Multi-view sampling for relevance feedback in image retrieval. In *Proceedings of the 18th IEEE International Conference on Pattern Recognition* (Hong Kong, China), 2, 881-884.

CHENG, H., HUA, K.A., AND VU, K. 2008. Leveraging user query log: toward improving image data clustering. In *Proceedings of the 7th ACM International Conference on Image and Video Retrieval* (Niagara Falls, ON, Canada), 27-36.

CHENG, E., JING, F, AND ZHANG, L. 2009. A unified relevance feedback framework for web image retrieval. *IEEE Transactions on Image Processing*, 18(6), 1350-1357.

CHIANG, C.-C., HSIEH, M.-H., HUNG, Y.-P., AND LEE G.C. 2005. Region filtering using color and texture features for image retrieval. In *Proceedings of the 4th ACM Conference on Image and Video Retrieval* (Singapore), 487-496.

CHOI, Y.S., AND NOH, J.S. 2004. Relevance feedback for content-based image retrieval using proximal support vector machine. In *Proceedings of the 2004 International Conference on Computational Science and Its Applications* (Assisi, Italy), 942-951.

CHOW, T.W.S., RAHMAN, M.K.S., AND WU, S. 2006. Content-based image retrieval by using tree-structured features and multi-layer self-organizing map. *Springer Journal of Pattern Analysis and Applications*, 9(1), 1-20.

COX, I.J., KILIAN, J., LEIGHTON, F.T., AND SHAMOON, T. 1997. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12), 1673-1687.

CRIMINISI, A., PÉREZ, P., AND TOYAMA, K. 2003. Object removal by exemplar-based inpainting. In *Proceedings of the 2003 IEEE Conference on Computer Vision and Pattern Recognition* (Madison, WI, USA), 2, 721-728.

DAGLI, C.K., RAJARAM, S., AND HUANG, T.S. 2006. Leveraging active learning for relevance feedback using an information theoretic diversity measure. In *Proceedings of the 5th ACM International Conference on Image and Video Retrieval* (Tempe, AZ, USA), 123-132.

DAS, G., RAY, S., AND WILSON, C. 2006. Feature re-weighting in content-based image retrieval. In *Proceedings of the 5th ACM International Conference on Image and Video Retrieval* (Tempe, AZ, USA), 193-200.

DAS, G., AND RAY, S. 2007. A comparison of relevance feedback strategies in CBIR. In *Proceedings of the 6th IEEE International Conference on Computer and Information Science* (Melbourne, VIC, Australia), 100-105.

DATTA, R., LI, J., AND WANG, J.Z. 2005. Content-based image retrieval: approaches and trends of the new age. In *Proceedings of the 7th ACM International Workshop on Multimedia Information Retrieval* (Singapore), 253-262.

DATTA, R., JOSHI, D., LI, J., AND WANG, J.Z. 2008. Image retrieval: ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2), 1-60.

DOLOC-MIHU A., AND RAGHAVAN, V.V. 2006. Score distribution approach to automatic kernel selection for image retrieval systems. In *Proceedings of the 16th International Symposium on Methodologies for Intelligent Systems* (Bari, Italy), 238-247.

DONG, A., AND BHANU, B. 2003a. A new semi-supervised EM algorithm for image retrieval. In *Proceedings of the 2003 IEEE Conference on Computer Vision and Pattern Recognition* (Madison, WI, USA), 2, 662-667.

DONG, A., AND BHANU, B. 2003b. Active concept learning for image retrieval in dynamic databases. In *Proceedings of the 9th IEEE International Conference on Computer Vision* (Nice, France), 90-95.

DOULAMIS, N. 2007. Optimal estimation of descriptor scales for multimedia retrieval. In *Proceedings of the 8th IEEE International Workshop on Image Analysis for Multimedia Interactive Services* (Santorini, Greece), 78-81.

DUYGULU, P., BARNARD, K., DE FREITAS, J.F.G., AND FORSYTH, D.A. 2002. Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. In *Proceedings of the 7th European Conference on Computer Vision* (Copenhagen, Denmark), 349-354.

EVANS, C. 2009. Notes on the OpenSURF library. Technical Report. University of Bristol.

EVERINGHAM, H., AND WINN, J. 2007. The Pascal VOC challenge 2007 development kit. Technical report. University of Leeds.

FAN, J., GAO, Y., LUO, H., AND JAIN R. 2008. Mining multilevel image semantics via hierarchical classification. *IEEE Transactions on Multimedia*, 10(2), 167-181.

FEI-FEI, L., FERGUS, R., AND PERONA, P. 2006. One-shot learning of object categories. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 28(4), 594-611.

FELLBAUM, C. 1998. WordNet: an electronic lexical database. MIT Press.

FERECATU, M., CRUCIANU, M., AND BOUJEMAA, N. 2004a. Retrieval of difficult image classes using SVM-based relevance feedback. In *Proceedings of the 6th ACM Workshop on Multimedia Information Retrieval* (New York, NY, USA), 23-30.

FERECATU, M., CRUCIANU, M., AND BOUJEMAA, N. 2004b. Sample selection strategies for relevance feedback in region-based image retrieval. In *Proceedings of the 5th Pacific Rim Conference on Multimedia* (Tokyo, Japan), 497-504.

FERECATU, M., BOUJEMAA, N., AND CRUCIANU, M. 2008. Semantic interactive image retrieval combining visual and conceptual content description. *ACM Multimedia Systems*, 13(5-6), 309-322.

FOO, J.J., ZOBEL, J., SINHA, R., AND TAHAGHOGHI, S.M.M. 2007a. Detection of near-duplicate images for web search. In *Proceedings of the 6th ACM Conference on Image and Video Retrieval* (Amsterdam, Netherlands), 557-564.

FOO, J.J., AND SINHA, R. 2007b. Pruning SIFT for scalable near-duplicate image matching. In *Proceedings of the 18th Australasian Database Conference* (Ballarat, VIC, Australia), 63-71.

FOO, J.J., SINHA, R., AND ZOBEL, J. 2007c. Discovery of image versions in large collections. In *Proceedings of the 13th International Multimedia Modeling Conference* (Singapore), 433-442.

FRANCO, A., LUMINI, A., AND MAIO, D. 2004. A new approach for relevance feedback through positive and negative samples. In *Proceedings of the 17th IEEE International Conference on Pattern Recognition* (Cambridge, UK), 4, 905-908.

FREUND, Y, AND SCHAPIRE, R.E. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119-139.

FU, Z., AND ROBLES-KELLY, A. 2009. An instance selection approach to multiple instance learning. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition* (Miami, Florida, USA), 911-918.

FUNG, C.C., AND CHUNG, K.-P. 2007. Establishing semantic relationship in inter-query learning for content-based image retrieval systems. In *Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining* (Nanjing, China), 498-506.

GEORGHIADES, A.S., BELHUMEUR, P.N., AND KRIEGMAN, D.J. 2001. From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6), 643-660.

GIACINTO, G., AND ROLI, F. 2002. Query shifting based on Bayesian decision theory for content-based image retrieval. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition* (Windsor, ON, Canada), 131-168.

GIACINTO, G., AND ROLI, F. 2003. Dissimilarity representation of images for relevance feedback in content-based image retrieval. In *Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition* (Leipzig, Germany), 195-207.

GIACINTO, G., AND ROLI, F. 2004. Nearest-prototype relevance feedback for content based image retrieval. In *Proceedings of the 17th IEEE International Conference on Pattern Recognition* (Cambridge, UK), 2, 989-992.

GIACINTO, G. 2007. A nearest-neighbor approach to relevance feedback in content based image retrieval. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval* (Amsterdam, Netherlands), 456-463.

GHOSH, P., DRELIE GELASCA, E., RAMAKRISHNAN, K.R., AND MANJUNATH, B.S. 2007. Duplicate image detection in large scale databases. *Advances in Intelligent Information Processing: Tools and Applications*, Eds. B. Chandra and C.A. Murthy, 149-166.

GOH, K.-S., LI, B., AND CHANG, E.Y. 2002. DynDex: a dynamic and non-metric space indexer. In *Proceedings of the 10th ACM International Conference on Multimedia* (Juan-les-Pins, France), 466-475.

GOH, K.-S., CHANG, E.Y., AND LAI, W.-C. 2004. Multimodal concept-dependent active learning for image retrieval. In *Proceedings of the 12th ACM International Conference on Multimedia* (New York, NY, USA), 564-571.

GONDRA, I., AND HEISTERKAMP, D.R. 2004. Learning in region-based image retrieval with generalized support vector machines. In *Proceedings of 2004 IEEE Conference on Computer Vision and Pattern Recognition Workshop* (Washington, DC, USA), 149-156.

GRIFFIN, G., HOLUB, A., AND PERONA, P. 2007. Caltech-256 object category dataset. Technical report, California Institute of Technology.

GRIGOROVA, A., DE NATALE, F.G.B., DAGLI, C.K., AND HUANG, T.S. 2007. Content-based image retrieval by feature adaptation and relevance feedback. *IEEE Transactions on Multimedia*, 9(6), 1183-1192.

GUAN, J., AND QIU, G. 2007a. Learning user intention in relevance feedback using optimization. In *Proceedings of the 9th ACM International Workshop on Multimedia Information Retrieval* (Augsburg, Germany), 41-50.

GUAN, J., AND QIU, G. 2007b. Modeling user feedback using a hierarchical graphical model for interactive image retrieval. In *Proceedings of the 8th Pacific Rim Conference on Multimedia* (Hong Kong, China), 18-29.

GUO, D., AND SIM, T. 2009. Color Me Right–Seamless Image Compositing. In *Proceedings of 13th International Conference on Computer Analysis of Images and Patterns* (Münster, Germany), 444-451.

HAAS, M., RIJSDAM., J., THOMEE, B. AND LEW, M.S. 2004. Relevance feedback methods in content based retrieval and video summarization. In *Proceedings of the 6th ACM International Workshop on Multimedia Information Retrieval* (New York, NY, USA), 151-156.

HAAS, M., OERLEMANS, A., AND LEW, M.S. 2005. Relevance feedback methods in content based retrieval and video summarization. In *Proceedings of the 2005 IEEE International Conference on Multimedia and Expo* (Amsterdam, Netherlands), 1038-1041.

HARNSOMBURANA, J., AND SHYU, C.-R. 2002. A hybrid tree approach for efficient image database retrieval with dynamic feedback. In *Proceedings of the 16th IEEE International Conference on Pattern Recognition* (Québec City, QC, Canada), 1, 263-266.

HAYS, J., AND EFROS, A.A. 2007. Scene completion using millions of photographs. *ACM Transactions on Graphics*, 26(3).

HE, X., MA, W.-Y., KING, O., LI, M., AND ZHANG, H.-J. 2002. Learning and inferring a semantic space from user's relevance feedback for image retrieval. In *Proceedings of the 10th ACM International Conference on Multimedia* (Juan-les-Pins, France), 343-346.

HE, X., AND NIYOGI, P. 2003. Locality preserving projections. *Advances in Neural Information Processing Systems*, 16, MIT Press.

HE, J., LI, M., ZHANG, H.-J., TONG, H., AND ZHANG, C. 2004a. Mean version space: a new active learning method for content-based image retrieval. In *Proceedings of the 6th ACM International Workshop on Multimedia Information Retrieval* (New York, NY, USA), 15-22.

HE, X. 2004b. Incremental semi-supervised subspace learning for image retrieval. In *Proceedings of the 12th ACM International Conference on Multimedia* (New York, NY, USA), 2-8.

HE, X., MA, W.-Y., AND ZHANG, H.-J. 2004c. Learning an image manifold for retrieval. In *Proceedings of the 12th ACM International Conference on Multimedia* (New York, NY, USA), 17-23.

HE, J., TONG, H., LI, M., MA, W.-Y., AND ZHANG, C. 2005. Multiple random walk and its application in content-based image retrieval. In *Proceedings of the 7th ACM International Workshop on Multimedia Information Retrieval* (Singapore), 151-158.

HE, X., MIN, W., CAI, D., AND ZHOU, K. 2007. Laplacian optimal design for image retrieval. In *Proceedings of the 30th ACM International Conference on Research and Development in Information Retrieval* (Amsterdam, Netherlands), 119-126.

HE, X., CAI, D., AND HAN, J. 2008. Learning a maximum margin subspace for image retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 20(2), 189-201.

HE, X. 2010. Laplacian regularized D-Optimal Design for active learning and its application to image retrieval. *IEEE Transactions on Image Processing*, 19(1), 254-263.

HEESCH, D.C., AND RÜGER, S. 2003. Relevance feedback for content-based image retrieval: what can three mouse clicks achieve? In *Proceedings of the 25th European conference on IR research* (Pisa, Italy), 545-558.

HEISTERKAMP, D.R., AND PENG, J. 2005. Kernel Vector Approximation Files for Relevance Feedback Retrieval in Large Image Databases. *Multimedia Tools and Applications*, 26(2), 175 - 189.

HOI, C.-H., AND LYU, M.R. 2004a. Group-based relevance feedback with support vector machine ensembles. In *Proceedings of the 17th IEEE International Conference on Pattern Recognition* (Cambridge, UK), 3, 874-877.

HOI, C.-H., CHAN, C., HUANG, K., LYU, M.R., AND KING, I. 2004b. Biased support vector machine for relevance feedback in image retrieval. In *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks* (Budapest, Hungary), 4, 3189-3194.

HOI, S.C.H., AND LYU, M.R. 2005. A semi-supervised active learning framework for image retrieval. In *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition* (San Diego, CA, USA), 2, 302-309.

HOI, S.C.H., LYU, M.R., AND JIN, R. 2006a. A unified log-based relevance feedback scheme for image retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 18(4), 509-524.

HOI, S.C.H., LIU, W., LYU, M.R., AND MA, W.-Y. 2006b. Learning distance metrics with contextual constraints for image retrieval. In *Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition* (New York, NY, USA), 2, 2072-2078.

HOI, S.C.H., JIN, R., ZU, J., AND LYU, M.R. 2009. Semisupervised SVM batch mode active learning with applications to image retrieval. *IEEE Transactions on Information Systems*, 27(3), article 16.

HOIEM, D., SUKTHANKAR, R., SCHNEIDERMAN, H., AND HUSTON, L. 2004. Object-based image retrieval using the statistical structure of images. In *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA), 2, 490-497.

HÖRSTER, E., LIENHART, R., AND SLANEY, M. 2007. Image retrieval on large-scale image databases. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval* (Amsterdam, Netherlands), 17-24.

HUANG, J., KUMAR, S.R., AND METRA, M. 1997. Combining supervised learning with color correlograms for content-based image retrieval. In *Proceedings of the 5th ACM International Conference on Multimedia* (Seattle, WA, USA), 325-334.

HUANG, R., LIU, Q LU, H., AND MA, S. 2002. Solving the small sample size problem of LDA. In *Proceedings of the 16th IEEE International Conference on Pattern Recognition* (Québec City, QC, Canada), 3, 29-32.

HUANG, X., CHEN, S.-C., AND SHYU, M.-L. 2003a. Incorporating real-valued multiple instance learning into relevance feedback for image retrieval. In *Pro-*

ceedings of the 2003 IEEE International Conference on Multimedia and Expo (Baltimore, MD, USA), 2, 321-324.

HUANG, X., CHEN, S.-C., SHYU, M.-L., AND ZHANG, C. 2003b. Mining high-level user concepts with multiple instance learning and relevance feedback for content-based image retrieval. *Mining Multimedia and Complex Data*, 2797, Eds. Springer, 50-67.

HUANG, S.-H., WU, Q.-J., AND LAI, S.-H. 2006. Improved AdaBoost-based image retrieval with relevance feedback via paired feature learning. *ACM Multimedia Systems*, 12(1), 14-26.

HUIJSMANS, D.P., AND SEBE, N. 2005. How to complete performance graphs in content-based image retrieval: add generality and normalize scope. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2), 245-251.

HUISKES, M.J. 2005. Aspect-based relevance learning for image retrieval. In *Proceedings of the 4th ACM International Conference on Image and Video Retrieval* (Singapore), 639-649.

HUISKES, M.J. 2006. Image searching and browsing by active aspect-based relevance learning. In *Proceedings of the 5th ACM International Conference on Image and Video Retrieval* (Tempe, AZ, USA), 211-220.

HUISKES, M.J., AND LEW, M.S. 2008a. Performance evaluation of relevance feedback methods. In *Proceedings of the 7th ACM International Conference on Image and Video Retrieval* (Niagara Falls, ON, Canada), 239-248.

HUISKES, M.J., AND LEW, M.S. 2008b. The MIR Flickr retrieval evaluation. In *Proceedings of the 10th ACM International Conference on Multimedia Information Retrieval* (Vancouver, BC, Canada), 39-43.

HUISKES, M.J., THOMEE, B., AND LEW, M.S. 2010. New trends and ideas in visual concept detection. In *Proceedings of the 11th ACM International Conference on Multimedia Information Retrieval* (Philadelphia, PA, USA), 527-536.

INDYK, P., AND MOTWANI, R. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the 30th ACM Symposium on Theory of Computing* (Dallas, TX, USA), 604-613.

ISHIKAWA, Y., SUBRAMANYA, R., AND FALOUTSOS, C. 1998. MindReader: querying databases through multiple examples. In *Proceedings of the 24th International Conference on Very Large Data Bases* (New York, NY, USA), 218-227.

JAIN, R. 2003. Experiential computing. *Communications of the ACM*, 46(7), 48-55.

JARRAH, K., AND GUAN, L. 2008. Content-based image retrieval via distributed databases. In *Proceedings of the 7th ACM Conference on Image and Video Retrieval* (Niagara Falls, ON, Canada), 389-394.

JI, R., AND YAO, H. 2007. Visual & textual fusion for region retrieval: from both fuzzy matching and Bayesian reasoning aspects. In *Proceedings of the 9th ACM*

International Workshop on Multimedia Information Retrieval (Augsburg, Germany), 159-168.

JI, R., YAO, H., LIU, S., WANG, J., AND XU, P. 2008. A novel retrieval refinement and interaction pattern by exploring result correlations for image retrieval. In *Proceedings of the 5th International Workshop on Adaptive Multimedia Retrieval: Retrieval, User, and Semantics* (Paris, France), 85-94.

JIANG, W., CHAN, K.L., LI, M., AND ZHANG, H.-J. 2005. Mapping low-level features to high-level semantic concepts in region-based image retrieval. In *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition* (San Diego, CA), 2, 244-249.

JIN, X., AND FRENCH, J.C. 2003. Improving image retrieval effectiveness via multiple queries. In *Proceedings of the 1st ACM International Workshop on Multimedia Databases* (New Orleans, LA, USA), 86-94.

JIN, X., FRENCH, J.C., AND MICHEL, J. 2006. Toward consistent evaluation of relevance feedback approaches in multimedia retrieval. In *Proceedings of the 3rd International Workshop on Adaptive Multimedia Retrieval: User, Context, and Feedback* (Glasgow, UK), 191-206.

JING, F., LI, M., ZHANG, L., ZHANG, H.-J., AND ZHANG, B. 2003. Learning in region-based image retrieval. In *Proceedings of the 10th ACM Conference on Image and Video Retrieval* (Juan-les-Pins, France), 199-204.

JING, F., LI, M., ZHANG, H.-J., AND ZHANG, B. 2004. Entropy-based active learning with support vector machines for content-based image retrieval. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo* (Taipei, Taiwan), 1, 85-88.

KARTHIK, S., AND JAWAHAR, C.V. 2006. Efficient region based indexing and retrieval for images with elastic bucket tries. In *Proceedings of the 18th IEEE International Conference on Pattern Recognition* (Hong Kong, China), 4, 169-172.

KÄSTER, T., PFEIFFER, M., AND BAUCKHAGE, C. 2006. Usability evaluation for image retrieval beyond desktop applications. In *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo* (Toronto, ON, Canada), 385-388.

KE, Y., SUKTHANKAR, R., AND HUSTON, L. 2004. Efficient near-duplicate and sub-image retrieval. In *Proceedings of the 12th ACM International Conference on Multimedia* (New York, NY, USA), 869-876.

KHERFI, M.L., ZIOU, D., AND BERNARDI, A. 2002. Learning from negative example in relevance feedback for content-based image retrieval. In *Proceedings of the 16th IEEE International Conference on Pattern Recognition* (Québec City, QC, Canada), 2, 933-936.

KHERFI, M.L., BRAHMI, D., AND ZIOU, D. 2004. Combining visual features with semantics for a more effective image retrieval. In *Proceedings of 17th IEEE International Conference on Pattern Recognition* (Cambridge, UK), 2, 961-964.

KIM, C. 2003. Content-based image copy detection. *Signal Processing: Image Communication*, 18(3), 169-184.

KO, B.C., AND BYUN, H. 2002a. Probabilistic neural networks supporting multi-class relevance feedback in region-based image retrieval. In *Proceedings of the 16th IEEE International Conference on Pattern Recognition* (Québec City, QC, Canada), 4, 138-141.

KO, B.C., AND BYUN, H. 2002b. Integrated region-based image retrieval using region's spatial relationships. In *Proceedings of the 16th IEEE International Conference on Pattern Recognition* (Québec City, QC, Canada), 1, 196-199.

KO, B.C., KWAK, S.Y., AND BYUN, H. 2004. SVM-based salient region(s) extraction method for image retrieval. In *Proceedings of the 17th IEEE International Conference on Pattern Recognition* (Cambridge, UK), 2, 977-980.

KOSKELA, M., LAAKSONEN, J., AND OJA, E. 2002. Implementing relevance feedback as convolutions of local neighborhoods on self-organizing maps. In *Proceedings of the 2002 International Conference on Artificial Neural Networks* (Madrid, Spain), 137-142.

KOZMA, L., KLAMI, A., AND KASKI, S. 2009. GaZIR: gaze-based zooming interface for image retrieval. In *Proceedings of the 2009 ACM International Conference on Multimodal Interfaces* (Cambridge, MA, USA), 305-312.

KUO, Y.-H., CHEN, K.-T., CHIANG, C.-H., AND HSU, W.H. 2009. Query expansion for hash-based image object retrieval. In *Proceedings of the 17th ACM International Conference on Multimedia* (Beijing, China), 65-74.

KUTICS, A., NAKAGAWA, A., TANAKA, K., YAMADA, M., SANBE, Y., AND OHTSUKA, S. 2003. Linking images and keywords for semantics-based image retrieval. In *Proceedings of the 2003 IEEE International Conference on Multimedia and Expo* (Baltimore, MD, USA), 1, 777-780.

KWATRA, V., SCHÖDL, A., ESSA, I., TURK, G., AND BOBICK, A. 2003. Graphcut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics*, 22(3), 277-286.

LA CASCIA, M., SETHI, S., AND SCLAROFF, S. 1998. Combining textual and visual cues for content-based image retrieval on the world wide web. In *Proceedings of the 1998 IEEE Workshop on Content-Based Access of Image and Video Libraries* (Santa Barbara, CA, USA), 24-28.

LAZEBNIK, S., SCHMID, C., AND PONCE, J. 2005. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8), 1265-1278.

LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

LEVINE, M.D. 1985. Vision in man and machine. McGraw-Hill, 574.

LEW, M.S., SEBE, N., DJERABA, C., AND JAIN, R. 2006. Content-based multimedia information retrieval: state of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2(1), 1-19.

LI, J., WANG, J.Z., AND WIEDERHOLD, G. 2000. IRM: integrated region matching for image retrieval. In *Proceedings of the 8th ACM International Conference on Multimedia* (Marina del Rey, CA, USA), 147-156.

LI, C.-J., AND HSU, C.-T. 2008. Image retrieval with relevance feedback based on graph-theoretic region correspondence estimation. *IEEE Transactions on Multimedia*, 10(3), 447-456.

LIM, J.-H., AND JIN, J.S. 2005. A structured learning framework for content-based image indexing and visual query. *ACM Multimedia Systems*, 10(4), 317-331.

LIN, Y.-Y., LIU, T.-L., AND CHEN H.-T. 2005. Semantic manifold learning for image retrieval. In *Proceedings of the 13th ACM International Conference on Multimedia* (Singapore), 249-258.

LIU, D., HUA, K.A., VU, K., AND YU, N. 2006a. Fast query point movement techniques with relevance feedback for content-based image retrieval. In *Proceedings of the 10th International Conference on Extending Database Technology* (Munich, Germany), 700-717.

LIU, Y., CHEN, X., ZHANG, C., AND SPRAGUE, A. 2006b. An interactive region-based image clustering and retrieval platform. In *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo* (Toronto, ON, Canada), 929-932.

LIU, J., LI, Z., LI, M., LU, H., AND MA, S. 2007a. Human behaviour consistent relevance feedback model for image retrieval. In *Proceedings of the 15th ACM International Conference on Multimedia* (Augsburg, Germany), 269-272.

LIU, R., WANG, Y., BABA, T., UEHARA, Y., MASUMOTO, D., AND NAGATA, S. 2007b. SVM-based active feedback in image retrieval using clustering and unlabeled data. In *Proceedings of the 12th International Conference on Computer Analysis of Images and Patterns* (Vienna, Austria), 954-961.

LIU, W., JIANG, W., AND CHANG, S.-F. 2008. Relevance aggregation projections for image retrieval. In *Proceedings of the 7th ACM International Conference on Image and Video Retrieval* (Niagara Falls, ON, Canada), 119-126.

LIU, X., CHENG, B., YAN, S., TANG, J., CHUA, T.S., AND JIN, H. 2009. Label to region by bi-layer sparsity priors. In *Proceedings of the 17th ACM International Conference on Multimedia* (Beijing, China), 115-124.

LOWE, D. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91-110.

LU, Y., ZHANG, H.-J., WENYIN, L., AND HU, C. 2003. Joint semantics and feature based image retrieval using relevance feedback. *IEEE Transactions on Multimedia*, 5(3), 339-347.

LU, C.-S., AND HSU, C.-Y. 2005. Geometric distortion-resilient image hashing scheme and its applications on copy detection and authentication. *ACM Multimedia Systems*, 11(2), 159-173.

LU, W., PAN, H., AND WU, J. 2006. Region-based semantic similarity propagation for image retrieval. *Advances in Multimedia Information Processing*, 4261, 1027-1036.

LU, Z., IP, H.H.S., AND HE, Q. 2009. Context-based multi-label image annotation. In *Proceedings of the 8th ACM International Conference on Image and Video Retrieval* (Santorini, Fira, Greece), article 30.

LUO, J., AND NASCIMENTO, M.A. 2004. Content-based sub-image retrieval using relevance feedback. In *Proceedings of the 2nd ACM International Workshop on Multimedia Databases* (Washington, DC, USA), 2-9.

MARAKAKIS, A., GALATSANOS, N., LIKAS, A., AND STAFYLOPATIS, A. 2008. A relevance feedback approach for content based image retrieval using Gaussian mixture models. In *Proceedings of the International Conference on Artificial Neural Networks* (Athens, Greece), 2, 84-93.

MARCHAND-MAILLET, S., AND WORRING, M. 2006. Benchmarking image and video retrieval: an overview. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval* (Santa Barbara, CA, USA), 297-300.

MARON, O., AND LOZANO-PÉREZ, T. 1998. A framework for multiple-instance learning. *Advances in Neural Information Processing Systems*, 10, MIT Press, 570-576.

MAVANDADI, S., AARABI, P., KHALEGHI, A., AND APPEL, R. 2006. Predictive dynamic user interfaces for interactive visual search. In *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo* (Toronto, ON, Canada), 381-384.

MENG, Y., CHANG, E., AND LI, B. 2003. Enhancing DPF for near-replica image recognition. In *Proceedings of the 2003 IEEE Conference on Computer Vision and Pattern Recognition* (Madison, WI, USA), 2, 416-423.

MIKOLAJCZYK, K., AND SCHMID, C. 2005. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), 1615-1630.

MUJA, M. AND LOWE, D. 2009. Fast approximate nearest neighbors with automatic algorithm configuration. In *2009 International Conference on Computer Vision Theory and Applications* (Lisbon, Portugal), 331-340.

MÜLLER, H., MARCHAND-MAILLET, S., AND PUN, T. 2002. The truth about Corel – evaluation in image retrieval. In *Proceedings of the 1st ACM International Conference on Image and Video Retrieval* (London, UK), 38-49.

MÜLLER, H., AND PUN, T. 2004. Learning from user behavior in image retrieval: application of market basket analysis. *Springer International Journal of Computer Vision*, 56(1-2), 65-77.

MÜLLER, H., DESELAERS, T., LEHMANN, T., CLOUGH, P., KIM, E., AND HERSH, W. 2007. Overview of the ImageCLEFmed 2006 medical retrieval and medical annotation tasks. In *Proceedings of the 7th Workshop of Cross-Language Evaluation Forum* (Alicante, Spain), 595-608.

MÜLLER, H., CLOUGH, P., DESELAERS, T., AND CAPUTO, B. 2010. ImageCLEF - experimental evaluation in visual information retrieval. Springer.

MUNEESAWANG, P., AND GUAN, L. 2003. Image retrieval with embedded sub-class information using Gaussian mixture models. In *Proceedings of the 2003 IEEE International Conference on Multimedia and Expo* (Baltimore, MD, USA), 2, 769-772.

MUNEESAWANG, P., AND GUAN, L. 2004. An interactive approach for CBIR using a network of radial basis functions. *IEEE Transactions on Multimedia*, 6(5), 703-716.

NAKAJIMA, S., KINOSHITA, S., AND TANAKA, K. 2003. Amplifying the differences between your positive samples and neighbors in image retrieval. In *Proceedings of the 2003 IEEE International Conference on Multimedia and Expo* (Baltimore, MD, USA), 1, 441-444.

NAKAZATO, M., AND HUANG, T.S. 2002. Extending image retrieval with group-oriented interface. In *Proceedings of the 2002 IEEE International Conference on Multimedia and Expo* (Lausanne, Switzerland), 1, 201-204.

NGUYEN, G.P., AND WORRING, M. 2005. Relevance feedback based saliency adaptation in CBIR. *ACM Multimedia Systems*, 10(6), 499-512.

NGUYEN, G.P., AND WORRING, M. 2006. Similarity learning via dissimilarity space in CBIR. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval* (Santa Barbara, CA, USA), 107-116.

NGUYEN, G.P., AND WORRING, M. 2008. Optimization of interactive visual-similarity-based search. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 4(1), 499-512.

NIKOLOPOULOS, S., ZAFEIRIOU, S., NIKOLAIDIS, N., AND PITAS, I. 2010. Image replica detection system utilizing R-trees and linear discriminant analysis. *Pattern Recognition*, 43(3), 636-649.

OH, S., CHUNG, M.G., AND SULL, S. 2004. Relevance feedback reinforced with semantics accumulation. In *Proceedings of the 3rd ACM International Conference on Image and Video Retrieval* (Dublin, Ireland), 448-454.

OLIVA, A., AND TORRALBA, A. 2001. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3), 145-175.

OLIVA, A., AND TORRALBA, A. 2006. Building the gist of a scene: the role of global image features in recognition. *Visual Perception, Progress in Brain Research*, 155, 23-36.

ORTEGA-BINDERBERGER, M., AND MEHROTRA, S. 2004. Relevance feedback techniques in the MARS image retrieval system. *ACM Multimedia Systems*, 9(6), 535-547.

PENG, X., AND KING, I. 2006a. Imbalanced learning in relevance feedback with biased minimax probability machine for image retrieval tasks. In *Proceedings of the 13th International Conference on Neural Information Processing* (Hong Kong, China), 342-351.

PENG, X., AND KING, I. 2006b. Biased minimax probability machine active learning for relevance feedback in content-based image retrieval. In *Proceedings of the 7th International Conference on Intelligent Data Engineering and Automated Learning* (Burgos, Spain), 953-960.

PÉREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *ACM Transactions on Graphics*, 22(3), 313-318.

PHILBIN, J., CHUM, O., ISARD, M., SIVIC, J., AND ZISSERMAN, A. 2007. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition* (Minneapolis, MN, USA), 1-8.

PICARD, D., CORD, M., AND REVEL, A. 2008. Image retrieval over networks: active learning using ant algorithm. *IEEE Transactions on Multimedia*, 10(7), 1356-1365.

PICKARD, R., GRASZYK, C., MANN, S., WACHMAN, J., PICKARD, L., AND CAMPBELL, L. 1995. VisTex databases. Technical report. MIT Media Laboratory.

QI, X., AND CHANG, R. 2007. Image retrieval using transaction-based and SVM-based learning in relevance feedback sessions. In *Proceedings of the 4th International Conference on Image Analysis and Recognition* (Montréal, QC, Canada), 638-649.

QIAN, F., LI, M., ZHANG, L., ZHANG, H.-J., AND ZHANG, B. 2002. Gaussian mixture model for relevance feedback in image retrieval. In *Proceedings of the 2002 IEEE International Conference on Multimedia and Expo* (Lausanne, Switzerland), 1, 229-232.

QIAN, F., ZHANG, B., AND LIN, F. 2003. Constructive learning algorithm-based RBF network for relevance feedback in image retrieval. In *Proceedings of the 2nd ACM International Conference on Image and Video Retrieval* (Urbana-Champaign, IL, USA), 133-138.

RAHMAN, M., BHATTACHARYA, P., AND DESAI, B.C. 2005. Probabilistic similarity measures in image databases with svm based categorization and relevance feedback. In *Proceedings of the 2nd International Conference on Image Analysis and Recognition* (Toronto, ON, Canada), 3656, 601-608.

RAHMANI, R., GOLDMAN, S.A., ZHANG, H., KRETTEK, J., AND FRITTS, J.E. 2005. Localized content based image retrieval. In *Proceedings of the 7th ACM International Workshop on Multimedia Information Retrieval* (Singapore), 227-236.

RAHMANI, R., GOLDMAN, S.A., ZHANG, H., CHOLLETI., S.R., AND FRITTS, J.E. 2008. Localized content-based image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11), 1902-1912.

RAMASWAMY, S., AND ROSE, K. 2009. Towards optimal indexing for relevance feedback in large image databases. *IEEE Transactions on Image Processing*, 18(12), 2780-2789.

RAO, Y., MUNDUR, P., AND YESHA, Y. 2006. Fuzzy SVM ensembles for relevance feedback in image retrieval. In *Proceedings of the 5th ACM International Conference on Image and Video Retrieval* (Tempe, AZ, USA), 350-359.

REGE, M., DONG, M., AND FOTOUHI, F. 2007. Building a user-centered semantic hierarchy in image databases. *ACM Multimedia Systems*, 12(4-5), 325-338.

REN, K., AND CALIC, J. 2009. FreeEye: interactive intuitive interface for large-scale image browsing. In *Proceedings of the 17th ACM International Conference on Multimedia* (Beijing, China), 757-760.

RO, Y.M., KIM, M., KANG, H.K., MANJUNATH, B.S., AND KIM, J. 2001. MPEG-7 homogeneous texture descriptor. *ETRI Journal*, 23(2), 41-51.

ROCCHIO, J.J. 1971. Relevance feedback in information retrieval. *The Smart Retrieval System: Experiments in Automatic Document Processing*, G. Salton (Ed.), Prentice Hall, 313-323.

ROWEIS, S.T., AND SAUL, L.K. 2000. Nonlinear dimensionality reduction by locally linear embedding, *Science*, 290(5500), 2323-2326.

ROYAL, M., CHANG, R., AND QI, X. 2007. Learning from relevance feedback sessions using a k-nearest-neighbor-based semantic repository. In *Proceedings of the 2007 IEEE International Conference on Multimedia and Expo* (Beijing, China), 1994-1997.

RUI, Y., HUANG, T.S., ORTEGA, M., AND MEHROTRA, S. 1998. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5), 644-655.

RUI, Y., AND HUANG, T.S. 2000. Optimized learning in image retrieval. In *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition* (Hilton Head, SC, USA), 1, 236-243.

SAHBI, H., AUDIBERT, J.-Y., AND KERIVEN, R. 2007. Graph-cut transducers for relevance feedback in content based image retrieval. In *Proceedings of the 11th IEEE International Conference on Computer Vision* (Rio de Janeiro, Brazil), 1-8.

SALTON, G., AND MCGILL, M. 1983. Introduction to modern information retrieval. McGraw-Hill.

SEBE, N., AND LEW, M.S. 2001. Color-based retrieval. *Pattern Recognition Letters*, 22(2), 223-230.

SHAH-HOSSEINI, A., AND KNAPP, G.M. 2006. Semantic image retrieval based on probabilistic latent semantic analysis. In *Proceedings of the 14th ACM International Conference on Multimedia* (Santa Barbara, CA, USA), 703-706.

SHYU, M., CHEN, S.-C., CHEN, M., ZHANG, C., AND SARINNAPAKORN, K. 2003. Image database retrieval utilizing affinity relationships. In *Proceedings of the 1st ACM International Workshop on Multimedia Databases* (New Orleans, LA, USA), 78-85.

SI, L., JIN, R., HOI, S.C.H., AND LYU, M.R. 2006. Collaborative image retrieval via regularized metric learning. *ACM Multimedia Systems*, 12(1), 34-44.

SIMONCELLI, E.P., AND PORTELLA, X. 1998. Texture characterization via joint statistics of wavelet coefficient magnitudes. In *Proceedings of the 5th International Conference on Image Processing* (Chicago, IL, USA), 1, 62-66.

SINGH, R., AND KOTHARI, R. 2003. Relevance feedback algorithm based on learning from labeled and unlabeled data. In *Proceedings of the 2003 IEEE International Conference on Multimedia and Expo* (Baltimore, MD, USA), 2, 433-436.

SMEATON, A.F., OVER, P., AND KRAAIJ, W. 2006. Evaluation campaigns and TRECVid. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval* (Santa Barbara, CA, USA), 1349-1380.

SMEULDERS, A.W.M., WORRING, M., SANTINI, S., GUPTA, A., AND JAIN, R. 2000. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12), 1349-1380.

STEJIĆ, Z., TAKAMA, Y., AND HIROTA, K. 2004. Small sample size performance of evolutionary algorithms for adaptive image retrieval. In *Proceedings of the 3rd ACM International Conference on Image and Video Retrieval* (Dublin, Ireland), 1968-1976.

SU, J., LIU, F., AND LUO, Z. 2005. Evolving optimal feature set by interactive reinforcement learning for image retrieval. In *Proceedings of the 2nd International Symposium on Neural Networks* (Chongqing, China), 2, 813-818.

SU, W.-T., CHU, W.-S., AND LIEN J.-J.J. 2006. Heuristic pre-clustering relevance feedback in region-based image retrieval. In *Proceedings of the 7th Asian Conference on Computer Vision* (Hyderabad, India), 2, 294-304.

SUN, Y., AND OZAWA, S. 2005. HIRBIR: a hierarchical approach to region-based image retrieval. *ACM Multimedia Systems*, 10(6), 559-569.

TANDON, P., NIGAM, P., PUDI, V., AND JAWAHAR, C.V. 2008. FISH: a practical system for fast interactive image search in huge databases. In *Proceedings of the 7th ACM International Conference on Image and Video Retrieval* (Niagara Falls, ON, Canada), 369-378.

TAO, J.-L., AND HUNG, Y.-P. 2002. A Bayesian method for content-based image retrieval by use of relevance feedback. In *Proceedings of the 5th International Conference on Recent Advances in Visual Information Systems* (Hsin Chu, Taiwan), 76-87.

TAO, D., AND TANG, X. 2004a. Orthogonal complement component analysis for positive samples in SVM based relevance feedback image retrieval. In *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA), 2, 586-591.

TAO, D., AND TANG, X. 2004b. Nonparametric discriminant analysis in relevance feedback for content-based image retrieval. In *Proceedings of the 17th IEEE International Conference on Pattern Recognition* (Cambridge, UK), 2, 1013-1016.

TAO, D., TANG, X., LI, X., AND RUI, Y. 2006a. Direct kernel biased discriminant analysis: a new content-based image retrieval relevance feedback algorithm. *IEEE Transactions on Multimedia*, 8(4), 716-727.

TAO, D., TANG, X., LI, X., AND WU, X. 2006b. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7), 1088-1099.

TENENBAUM, J.B., SILVA, V.D., AND LANGFORD, J.C. 2000. A global geometric framework for nonlinear dimensionality reduction, *Science*, 290(5500), 2319-2323.

TEŠIĆ, J., AND MANJUNATH, B.S. 2003. Nearest neighbor search for relevance feedback. In *Proceedings of the 2003 IEEE Conference on Computer Vision and Pattern Recognition* (Madison, WI, USA), 2, 643-648.

THERRIEN, C.W. 1989. Decision, estimation and classification. John Wiley & Sons.

THOMEE, B., HUISKES, M.J., BAKKER, E.M., AND LEW, M.S. 2007a. Visual information retrieval using synthesized imagery. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval* (Amsterdam, Netherlands), 127-130.

THOMEE, B., HUISKES, M.J., BAKKER, E.M., AND LEW, M.S. 2007b. An artificial imagination for interactive search. In *Proceedings of the IEEE International Workshop on Human-Computer Interaction* (Rio de Janeiro, Brazil), 19-28.

THOMEE, B., HUISKES, M.J., BAKKER, E.M., AND LEW, M.S. 2008a. Using an artificial imagination for texture retrieval. In *Proceedings of the 19th IEEE International Conference on Pattern Recognition* (Tampa, FL, USA), 1-4.

THOMEE, B., HUISKES, M.J., BAKKER, E.M., AND LEW, M.S. 2008b. Large scale image copy detection evaluation. In *Proceedings of the 10th ACM International Conference on Multimedia Information Retrieval* (Vancouver, BC, Canada), 59-66.

THOMEE, B., HUISKES, M.J., BAKKER, E.M., AND LEW, M.S. 2009a. An exploration-based interface for interactive image retrieval. In *Proceedings of the 6th IEEE International Symposium on Image and Signal Processing* (Salzburg, Austria), 192-197.

THOMEE, B., HUISKES, M.J., BAKKER, E.M., AND LEW, M.S. 2009b. Deep exploration for experiential image retrieval. In *Proceedings of the 17th ACM International Conference on Multimedia* (Beijing, China), 673-676.

THOMEE, B., HUISKES, M.J., BAKKER, E.M., AND LEW, M.S. 2009c. Combining visual exploration and searching for interactive texture retrieval. In *Proceedings of the 21st Benelux Conference on Artificial Intelligence* (Eindhoven, Netherlands), paper 16.

THOMEE, B., BAKKER, E.M., AND LEW, M.S. 2010. TOP-SURF: a visual words toolkit. Accepted for publication in *Proceedings of the 18th ACM International Conference on Multimedia* (Firenze, Italy).

TIEU, K., AND VIOLA, P. 2004. Boosting image retrieval. *Springer International Journal of Computer Vision*, 56(1-2), 17-36.

TONG, S, AND CHANG, E. 2001. Support vector machine active learning for image retrieval. In *Proceedings of the 9th ACM International Conference on Multimedia* (Ottawa, ON, Canada), 107-118.

TONG, H., HE, J., LI, M., MA, W.-Y., ZHANG, C., AND ZHANG, H.-J. 2005. A unified optimization based learning method for image retrieval. In *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition* (San Diego, CA, USA), 2, 230-235.

TORRES, J.M., HUTCHISON, D., AND REIS, L.P. 2007. Semantic image retrieval using region-based relevance feedback. In *Proceedings of the 4th International Workshop on Adaptive Multimedia Retrieval: User, Context, and Feedback* (Geneva, Switzerland), 192-206.

TRAN, D.A., PAMIDIMUKKALA, S.R., AND NGUYEN, P. 2008. Relevance-feedback image retrieval based on multiple-instance learning. In *Proceedings of the 7th*

IEEE International Conference on Computer and Information Science (Paris, France), 597-602.

TSIKRIKA, T., DIOU, C., VRIES, A.P. DE, AND DELOPOULOS, A. 2009. Image annotation using clickthrough data. In *Proceedings of the 8th ACM International Conference on Image and Video Retrieval* (Santorini, Fira, Greece), article 14.

TU, Y., LI, G., AND DAI, H. 2008. Integrating local one-class classifiers for image retrieval. In *Proceedings of the 2nd International Conference on Advanced Data Mining and Applications* (Xi'an, China), 213-222.

URBAN, J., AND JOSE, J.M. 2006a. Adaptive image retrieval using a graph model for semantic feature integration. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval* (Santa Barbara, CA, USA), 117-126.

URBAN, J., JOSE, J.M., AND RIJSBERGEN, C.J. 2006b. An adaptive technique for content-based image retrieval. *Springer Multimedia Tools and Applications*, 31(1), 1-28.

URBAN, J., AND JOSE, J.M. 2007. Evaluating a workspace's usefulness for image retrieval. *ACM Multimedia Systems*, 12(4-5), 355-373.

WANG, J.Z., LI, J., AND WIEDERHOLD, G. 2001. SIMPLIcity: semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9), 947-963.

WANG, T., RUI, Y., HU, S.-M., AND SUN, J.-G. 2003a. Adaptive tree similarity learning for image retrieval. *ACM Multimedia Systems*, 9(2), 131-143.

WANG, L., AND CHAN, K.L. 2003b. A dynamic sub-vector weighting scheme for image retrieval with relevance feedback. *Springer Pattern Analysis and Applications*, 6(3), 212-223.

WANG, L., GAO, Y., CHAN, K.L., XUE, P., AND YAU, W.-Y. 2005a. Retrieval with knowledge-driven kernel design: an approach to improving SVM-based CBIR with relevance feedback. In *Proceedings of the 10th IEEE International Conference on Computer Vision* (Beijing, China), 2, 1355-1362.

WANG, L., LI, X., XUE, P., AND CHAN, K.L. 2005b. A novel framework for SVM-based image retrieval on large databases. In *Proceedings of the 13th ACM International Conference on Multimedia* (Singapore), 487-490.

WANG, X.-J., MA, W.-Y., ZHANG, L., AND LI, X. 2005c. Multi-graph enabled active learning for multimodal web image retrieval. In *Proceedings of the 7th ACM International Workshop on Multimedia Information Retrieval* (Singapore), 65-72.

WANG, B., LI, Z., LI, M., AND MA, W.-Y. 2006. Large-scale duplicate detection for web image search. In *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo* (Toronto, ON, Canada), 353-356.

WANG, X., MCKENNA, S.J., AND HAN, J. 2009. High-entropy layouts for content-based browsing and retrieval. In *Proceedings of the 8th ACM International Conference on Image and Video Retrieval* (Santorini, Fira, Greece), article 16.

WILCZKOWIAK, M., BROWSTOW, G.J., TORDOFF, B., AND CIPOLLA, R. 2005. Hole filling through photomontage. In *Proceedings of the 16th British Machine Vision Conference* (Oxford, UK), 492-501.

WU, H., LU, H., AND MA, S. 2002a. The role of sample distribution in relevance feedback for content based image retrieval. In *Proceedings of the 2002 IEEE International Conference on Multimedia and Expo* (Lausanne, Switzerland), 225-228.

WU, Y., AND ZHANG, A. 2002b. A feature re-weighting approach for relevance feedback in image retrieval. In *Proceedings of the 9th IEEE International Conference on Image Processing* (Rochester, NY, USA), 2, 581-584.

WU, H., LU, H., AND MA, S. 2003a. Multilevel relevance judgment, loss function, and performance measure in image retrieval. In *Proceedings of the 2nd ACM International Conference on Image and Video Retrieval* (Urbana-Champaign, IL, USA), 409-414.

WU, H., LU, H., AND MA, S. 2003b. A practical SVM-based algorithm for ordinal regression in image retrieval. In *Proceedings of the 11th ACM International Conference on Multimedia* (Berkeley, CA, USA), 612-621.

WU, H., LU, H., AND MA, S. 2004a. WillHunter: interactive image retrieval with multilevel relevance measurement. In *Proceedings of the 17th IEEE International Conference on Pattern Recognition* (Cambridge, UK), 2, 1009-1012.

WU, Y., AND ZHANG, A. 2004b. Interactive pattern analysis for relevance feedback in multimedia information retrieval. *ACM Multimedia Systems*, 10(1), 41-55.

WU, Y., AND ZHANG, A. 2004c. PatternQuest: learning patterns of interest using relevance feedback in multimedia information retrieval. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo* (Taipei, Taiwan), 1, 261-264.

WU, K., YAP, K.-H., AND CHAU, L.-P. 2006. Region-based image retrieval using radial basis function network. In *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo* (Toronto, ON, Canada), 1777-1780.

XIE, H., AND ORTEGA, A. 2004. An user preference information based kernel for SVM active learning in content-based image retrieval. In *Proceedings of the 6th ACM International Workshop on Multimedia Information Retrieval* (New York, NY, USA), 1-6.

XIE, H., ANDREU, V., AND ORTEGA, A. 2006. Quantization-based probabilistic feature modeling for kernel design in content-based image retrieval. In *Proceed-*

ings of the 8th ACM International Workshop on Multimedia Information Retrieval (Santa Barbara, CA, USA), 23-32.

YANG, J., LI, Q., AND ZHUANG, Y. 2002. Image retrieval and relevance feedback using peer indexing. In *Proceedings of the 2002 IEEE International Conference on Multimedia and Expo* (Lausanne, Switzerland), 2, 409-412.

YANG, J., LI, Q., AND ZHUANG, Y. 2004. Towards data-adaptive and user-adaptive image retrieval by peer indexing. *Springer International Journal of Computer Vision*, 56(1-2), 47-63.

YANG, C., DONG, M., AND FOTOUHI, F. 2005. Semantic feedback for interactive image retrieval. In *Proceedings of the 13th ACM International Conference on Multimedia* (Singapore), 415-418.

YANG, H., WANG, Q., AND HE, Z. 2008. Randomized sub-vectors hashing for high-dimensional image feature matching. In *Proceedings of the 16th ACM International Conference on Multimedia* (Vancouver, BC, Canada), 705-708.

YANG, X., ZHU, Q., AND CHENG, K.-T. 2009. Near-duplicate detection for images and videos. In *Proceedings of the 1st ACM Workshop on Large-scale Multimedia Retrieval and Mining* (Beijing, China), 73-80.

YIN, P.-Y., BHANU, B., CHANG, K.-C., AND DONG, A. 2002. Improving retrieval performance by long-term relevance information. In *Proceedings of the 16th IEEE International Conference on Pattern Recognition* (Québec City, QC, Canada), 3, 533-536.

YIN, P.-Y., BHANU, B., CHANG, K.-C., AND DONG, A. 2005. Integrating relevance feedback techniques for image retrieval using reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), 1536-1551.

YIN, P.-Y., BHANU, B., CHANG, K.-C., AND DONG, A. 2008. Long-term cross-session relevance feedback using virtual features. *IEEE Transactions on Knowledge and Data Engineering*, 20(3), 352-368.

YOON, J., AND JAYANT, N. 2002. Prefetching for content-based image retrieval. In *Proceedings of the 2002 IEEE International Conference on Multimedia and Expo* (Lausanne, Switzerland), 2, 413-416.

YOSHIZAWA, T., AND SCHWEITZER, H. 2004. Long-term learning of semantic grouping from relevance-feedback. In *Proceedings of the 6th ACM International Workshop on Multimedia Information Retrieval* (New York, NY, USA), 165-172.

YU, J., AND TIAN, Q. 2006. Learning image manifolds by semantic subspace projection. In *Proceedings of the 14th ACM International Conference on Multimedia* (Santa Barbara, CA, USA), 297-306.

YU, N., VU, K., AND HUA, K.A. 2007. An in-memory relevance feedback technique for high-performance image retrieval systems. In *Proceedings of the 6th ACM*

International Conference on Image and Video Retrieval (Amsterdam, Netherlands), 9-16.

ZAVESKY, E., CHANG, S.-F., AND YANG, C.-C. 2008. Visual islands: intuitive browsing of visual search results. In *Proceedings of the 7th ACM International Conference on Image and Video Retrieval* (Niagara Falls, ON, Canada), 617-626.

ZHANG, L., LIN, F., AND ZHANG, B. 2001. Support vector machine learning for image retrieval. In *Proceedings of the 2001 IEEE International Conference on Image Processing*, 2, 721-724.

ZHANG, H.-J., CHEN, Z., LI, M., AND SU, Z. 2003. Relevance feedback and learning in content-based image search. *Springer Journal of World Wide Web*, 6(2), 131-155.

ZHANG, R., AND ZHANG, Z. 2004a. Hidden semantic concept discovery in region based image retrieval. In *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition* (Cambridge, UK), 2, 996-1001.

ZHANG, R., AND ZHANG, Z. 2004b. Stretching Bayesian learning in the relevance feedback of image retrieval. In *Proceedings of the 8th European Conference on Computer Vision* (Prague, Czech Republic), 3, 996-1001.

ZHANG, R., AND ZHANG, Z. 2005a. FAST: toward more effective and efficient image retrieval. *ACM Multimedia Systems*, 10(6), 529-543.

ZHANG, C., CHEN, X., CHEN, M., CHEN, S.-C., AND SHYU, M.-L. 2005b. A multiple instance learning approach for content based image retrieval. In *Proceedings of the 2005 IEEE International Conference on Multimedia and Expo* (Amsterdam, Netherlands), 1142-1145.

ZHANG, C., AND CHEN, X. 2005c. Region-based image clustering and retrieval using multiple instance learning. In *Proceedings of the 4th ACM International Conference on Image and Video Retrieval* (Singapore), 194-204.

ZHANG, J., AND YE, L. 2007a. An unified framework based on p-norm for feature aggregation in content-based image retrieval. In *Proceedings of the 9th IEEE International Symposium on Multimedia* (Taichung, Taiwan), 195-201.

ZHANG, N., AND GUAN, L. 2007b. Graph cuts in content-based image classification and retrieval with relevance feedback. *Advances in Multimedia Information Processing*, 4810, 30-39.

ZHANG, X., CHENG, J., LU, H., AND MA, S. 2008. Selective sampling based on dynamic certainty propagation for image retrieval. In *Proceedings of the 14th International Multimedia Modeling Conference* (Kyoto, Japan), 425-435.

ZHANG, L., CHEN, C., CHEN, W., BU, J., CAI, D., AND HE, X. 2009a. Convex experimental design using manifold structure for image retrieval. In *Proceedings of the 17th ACM International Conference on Multimedia* (Beijing, China), 45-54.

ZHANG, X., CHENG, J., XU, C., LU, H. AND MA, S. 2009b. Multi-view multi-label active learning for image classification. In *Proceedings of the 2009 IEEE International Conference on Multimedia and Expo* (New York, NY, USA), 258-261.

ZHANG, J., AND YE, L. 2009c. Image retrieval using noisy query. In *Proceedings of the 2009 IEEE International Conference on Multimedia and Expo* (New York, NY, USA), 866-869.

ZHANG, J., AND YE, L. 2009d. Content based image retrieval using unclean positive examples. *IEEE Transactions on Image Processing*, 18(10), 2370-2375.

ZHOU, X.S., AND HUANG, T.S. 2001. Small sample learning during multimedia retrieval using BiasMap. In *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition* (Kauai, HI, USA), 1, 11-17.

ZHOU, X., ZHANG, Q., LIN, L., DENG, A., AND WU, G. 2003a. Image retrieval by fuzzy clustering of relevance feedback records. In *Proceedings of the 2003 IEEE International Conference on Multimedia and Expo* (Baltimore, MD, USA), 2, 305-308.

ZHOU, X., ZHANG, Q., ZHANG, L., LIU, L., AND SHI, B. 2003b. An image retrieval method based on collaborative filtering. *Intelligent Data Engineering and Automated Learning*, 2690, Eds. Springer, 1024-1031.

ZHOU, X.S., AND HUANG T.S. 2003c. Relevance feedback in image retrieval: a comprehensive review. *ACM Multimedia Systems*, 8(6), 536-544.

ZHOU, X.S., GARG, A., AND HUANG, T.S. 2004. A discussion of nonlinear variants of biased discriminants for interactive image retrieval. In *Proceedings of the 3rd ACM International Conference on Image and Video Retrieval* (Dublin, Ireland), 1948-1959.

ZHOU, Z.-H., CHEN, K.-J., AND DAI, H.-B. 2006. Enhancing relevance feedback in image retrieval using unlabeled data. *ACM Transactions on Information Systems*, 24(2), 219-244.

Nederlandse samenvatting

In het huidige digitale tijdperk is de computer niet meer weg te denken en is deze zelfs superieur aan de mens op allerlei gebieden, bijvoorbeeld in het oplossen van ingewikkelde wiskundige materie of het voorspellen van het weer. Echter op andere gebieden is de computer helemaal niet zoveel slimmer, alhoewel hij veel van zijn tekortkomingen kan compenseren met brute rekenkracht. Met name op het vlak wat de mens zo uniek maakt, het analyseren van materie vanuit een hoger perspectief en het vormen van relaties tussen velerlij zaken, legt de computer het vaak af qua intelligentie en capaciteit. Eén van deze gebieden betreft het zoeken naar multimediale informatie, het onderzoeksveld waar dit proefschrift zich op richt.

Nu de tendens is om steeds meer informatie te verplaatsen naar persoonlijke computers of het internet, komt er een extra dimensie bij kijken als het gaat om het (terug)vinden van deze informatie. Zoekmachines zoals Google en Yahoo! tonen aan dat dit goed lukt voor textuele informatie. Echter, als het gaat om beeldmateriaal zijn de resultaten wisselvallig. Het spreekwoord “een beeld zegt meer dan duizend woorden” is kenmerkend voor het probleem, zeker omdat deze woorden ook variëren per persoon die naar het beeld kijkt. Wat door een persoon als een “vakantiekiekje van een berg” wordt gezien, kan door een ander worden beschreven als “landschap van IJsland”, en door een derde als “de Eyjafjallajökull vulkaan op het punt van uitbarsten”. Om mensen naar foto’s te kunnen laten kunnen zoeken, zal een computer dus met allerlei mogelijke omschrijvingen van de foto rekening moeten houden.

In dit proefschrift doen wij onderzoek naar hoe de computer de inhoud van beelden zodanig kan interpreteren, dat het de juiste beelden kan bepalen waarnaar de gebruiker op zoek is. Deze zoektocht kan eventueel plaatsvinden in combinatie met een terugkoppelingsfase, waarbij de gebruiker aangeeft welke van de getoonde beelden zij relevant danwel irrelevant vindt in vergelijking met het door haar gezochtte beeld. Ook presenteren wij onderzoek naar methodes die de computer beelden laat genereren die zich richten op bepaalde beeldkenmerken waarin de gebruiker specifiek geïnteresseerd lijkt te zijn, een nieuw paradigma dat wij kunstmatige verbeelding noemen. Verder omvat ons werk onderzoek naar verbeterde manieren om makkelijk in een beeldencollectie te zoeken en vergelijken wij verscheidene methoden die als doel hebben de (bijna-)kopiën van een beeld te onderscheiden van alle andere beelden in een verzameling. Als laatste lichten wij de sluier op van een tweetal projecten die zich momenteel nog in de ontwikkelingsfase bevinden. In de volgende secties zullen wij de hoofdlijnen van het proefschift bespreken.

In hoofdstuk 1 introduceren wij ons vakgebied, waarbij wij de huidige standaarden, definities en terminologie bespreken. Wij tonen voorbeelden van de beeldenverzamelingen die doorgaans door andere onderzoekers in ons vakgebied gebruikt worden, evenals populaire technieken om beelden te representeren vanuit het oogpunt van de computer. Ook introduceren wij evaluatiemethoden die gebruikt worden voor het bepalen van de effectiviteit van een nieuwe ontwikkeling, en illustreren wij deze methoden aan de hand van een fictieve onderzoekssituatie.

In hoofdstuk 2 brengen wij het onderzoekslandschap in kaart van het vakgebied dat zich bezig houdt met het interactief zoeken naar beelden. Het rapport bestaat uit een analyse van meer dan 200 artikelen die tussen 2002-2010 gepubliceerd zijn en biedt inzicht in welke richtingen het vakgebied zich momenteel aan het ontwikkelen is. Verder dient dit rapport als doel om de context weer te geven waarin ons eigen onderzoek zich bevindt.

Het nieuwe paradigma dat wij kunstmatige verbeelding noemen wordt beproken in hoofdstuk 3. Kunstmatige verbeelding biedt de zoekmachine de mogelijkheid de verbeelding aan te spreken met als doel om beelden te synthetiseren die idealiter lijken op datgene waar de gebruiker naar op zoek is. Wij presenteren een methode, geïnspireerd door evolutionaire algorithmen, voor het genereren van texturen, die zich richten op bepaalde beeldkenmerken waarin de gebruiker geïnteresseerd lijkt te zijn. Het vraagstuk is of gebruiker sneller de voor haar interessante beelden kan vinden als deze gesynthetiseerde beelden door de zoekmachine aangewend kunnen worden tijdens de zoektocht. Onze hypothese is dat positieve danwel negatieve feedback van de gebruiker op deze beelden de computer betere informatie verschaft over haar wensen dan als zij feedback zou hebben gegeven op een regulier plaatje uit de beeldencollectie.

Wij zien verscheidene parallellen tussen kunstmatige verbeelding en kunstmatige intelligentie, waarbij de laatste een enorme ontwikkeling heeft doorgemaakt van primitieve intelligentie tot het niveau van vandaag, met als ultiem doel het niveau van menselijke intelligentie te evenaren of zelfs te overtreffen. In dit opzicht zien wij ook de ontwikkeling van kunstmatige verbeelding, waarbij over enkele jaren ons huidige werk misschien als primitief ervaren zal worden, maar waarin het wellicht ook aan de basis staat van een ontwikkelingsrevolutie, die uiteindelijk zal resulteren in kunstmatige verbeelding die in staat is menselijke verbeelding te evenaren.

Een van de grote vraagstukken, zoals gesteld door onderzoekers in ons vakgebied, betreft de noodzaak voor experientiële systemen die de gebruiker in staat stellen inzicht te krijgen in mediacollecties en haar de mogelijkheid bieden deze te exploreren. In hoofdstuk 4 presenteren wij een experientieel systeem, welke de gebruiker via een intuitieve interface door de collectie laat grasduinen. Een nieuw navigatiemechanisme, *deep exploration* genoemd, stelt de gebruiker in staat eenvoudig beelden te vinden die zich op diverse plekken in de collectie bevinden. Het zoeksysteem vindt beelden die lijken op een bepaalde zoekopdracht door de kleur en texturen te vergelijken met alle plaatjes in de collectie. Als de gebruiker feedback geeft poogt het systeem automatisch te distilleren in wat voor mate de gebruiker gericht is op bepaalde kleuren en op bepaalde texturen, en gebruikt het deze informatie om een betere selectie beelden aan de gebruiker te kunnen retourneren.

In de latere hoofdstukken verleggen wij onze focus naar collecties die miljoenen beelden kunnen bevatten. Met zo veel beelden is het zoeken naar de plaatjes waar de gebruiker naar op zoek is een echte opgave, o.a. vanuit het oogpunt van vereiste rekenkracht, opslag- en geheugencapaciteit, en acceptabele precisie in de gevonden resultaten. In hoofdstuk 5 beschrijven wij een techniek, genoemd TOP-SURF, om op een compacte manier beelden door de computer te laten representeren. TOP-SURF combineert de techniek die opvallende details vindt in beelden met die gebaseerd op visuele woorden. Sterke punten van TOP-SURF zijn dat snel beelden met elkaar vergeleken kunnen worden en dat de representatiegrootte afgestemd kan worden op het beschikbare geheugen en/of gewenste precisie in de resultaten.

In hoofdstuk 6 vergelijken we diverse kopieherkenningstechnieken met elkaar, gebruikmakend van een database die in totaal meer dan 4 miljoen beelden bevat. De kopiën zijn gemaakt door originelen aan te passen door middel van diverse transformaties, bijvoorbeeld door ze groter of kleiner te maken of er een copyright logo op te plaatsen. In de techniekvergelijking richten wij ons op de hoeveelheid rekenkracht die benodigd is, hoeveel geheugen de methodes gebruiken en hoe snel en hoe goed ze de kopiën kunnen onderscheiden van alle andere beelden.

De experimenten uitgevoerd in de hoofdstukken 5 en 6 hebben gebruikgemaakt van beelden die verkregen zijn van het internet. In appendix A bespreken wij onder andere de structuur van onze beeldensnuffelaar, die het internet afstruint en alle beelden die hij tegenkomt in een verzameling opslaat. Tevens presenteren wij een bètaversie van onze ‘opmerkelijke beeldenzoekmachine’, die als doel heeft de gebruiker beelden voor te schotelen die voor haar opmerkelijk zijn. Voor een filmfanaat zijn dat bijvoorbeeld alle beelden die met films te maken hebben, terwijl dat voor een atleet vooral beelden zullen zijn die met sport te maken hebben. Tegelijkertijd integreren wij onze TOP-SURF kopieherkenningstechniek om de getoonde zoekresultaten te diversificeren, door kopiën te groeperen en zodoende meerdere verschillende beelden te kunnen tonen. Onze huidige zoekmachine is nog niet gereed en vooral de personalisatie van de zoekresultaten vereist nog enige aandacht, maar desalniettemin is het vooruitzicht van onze zoekmachine rooskleurig.

Als laatste onderwerp presenteren wij in appendix B de bètaversie van onze ‘retoucheer beeldenzoekmachine’, welke is gebaseerd op de principes van scenevoltooiing. De zoekmachine biedt de gebruiker de mogelijkheid om op interactieve wijzen ongewenste beeldelementen te verwijderen en deze elementen te laten vervangen door andere inhoud. Deze vervangende inhoud is niet alleen zodanig geconstrueerd dat het naadloos in het originele beeld past, maar zorgt er ook voor dat de algehele inhoud nog steeds als realistisch ervaren wordt. Oftewel, als er in een foto van Rome een auto van de voorgrond wordt weggehaald, zal deze niet vervangen worden door een pinguïn, maar eerder door bijvoorbeeld een fontein. Onze veronderstelling is dat de ongewenste beeldelementen een negatieve impact hebben op de zoekresultaten, en wij denken dat als de gebruiker met deze geretoucheerde beelden een zoekactie start naar andere beelden die er op lijken, de zoekresultaten meer beelden zullen bevatten die van interesse zijn voor de gebruiker, dan als de gebruiker gezocht zou hebben met het onaangepaste beeld. Omdat de zoekmachine niet helemaal gereed is hebben wij onze hypothese nog niet kunnen testen, maar de gebruikte technieken zijn veelbelovend en naderen voltooiing. Derhalve hopen wij op korte termijn met een volledig functionele zoekmachine aan de slag te kunnen.

Acknowledgements

I would like to sincerely thank Michael Lew, Mark Huiskes, Erwin Bakker, Alan Smeaton, Kiyoharu Aizawa, Toshihiko Yamasaki and Hyowon Kim for the excellent discussions and critical suggestions. I would also like to thank Leiden University, NWO, ASCI and the students that participated in my user experiments for their assistance. Finally, many thanks to my parents for their endless support.

Curriculum vitae

Personal details

Name:	Bart Thomée
Date of birth:	18 February 1981
Place of birth:	Delft, Netherlands
Nationality:	Dutch, British

Education

PhD in Computer Science at Leiden University, Netherlands	2006-2010
MSc in Computer Science at Leiden University, Netherlands	1999-2006
VWO at Grotius College, Delft, Netherlands	1993-1999

Academic biography

Bart Thomée received the MSc degree in computer science from Leiden University in 2006. During his studies, he expanded his skill set by actively pursuing extracurricular coursework related to image, video and audio processing. His strong interest in foreign affairs stimulated him to study a year in the USA, attending Winston-Salem State University and The University of North Carolina at Greensboro. Besides being a native speaker of Dutch and English, Bart can converse in German, French and Spanish, and has basic knowledge of Mandarin Chinese and Japanese. This thesis is the final step towards obtaining his PhD from Leiden University under supervision of Dr. M.S. Lew. To expand his horizon and get acquainted with other researchers in his field, Bart visited the research labs of Prof. Dr. A.F. Smeaton at Dublin City University and Prof. Dr. K. Aizawa at The University of Tokyo. Bart's research resulted in 16 peer-reviewed publications, of which one received the 'outstanding paper citation' award. Furthermore, he lectured classes and workshops, supervised master's students and served as a reviewer for journals and conferences. Bart's research interests lie mainly in the area of multimedia: information retrieval, video summarization, computer networks, and streaming media.