J.C. van Gemert

April 2003

UNIVERSITEIT VAN AMSTERDAM

Retrieving Images as Text

Combining Images and Text for Information Retrieval

Master's thesis Artificial Intelligence, specialization Logic.

Under supervision of Dr. Jan-Mark Geusebroek

April 2003

by

Johannes Christianus van Gemert

Intelligent Sensory Information Systems
Informatics Institute
Faculty of Science
Universiteit van Amsterdam

aan ons pap, en ons mam

Abstract

Both interpretations of the title *Retrieving Images as Text* are considered in this thesis. We use text retrieval methods for content based image retrieval and we introduce a method for retrieving and combining images and text. A new method for image retrieval is introduced using Gaussian color scale space. It uses local features to describe small pieces of images where each piece is labeled. Such a label is denoted a 'word' of the image. LSI is applied on the words and yields good general results on a dataset for content based image retrieval and particularly on occluded images and camera rotated images, demonstrating the locality of the features. Images and text are combined by concatenating the words in an image to the textual words describing the image. When LSI is applied on this combined semantic space, relations are learned between text and text, image and image and text and image. Some examples demonstrate the usefulness of the method.

Keywords: Information Retrieval, Content Based Image Retrieval, Digital Libraries, Multimodal Retrieval, Combining Images and Text, Image Semantics, Local Image Color Structure.

Contents

Chapter 1

Introduction

It is said that an image contains more information than a thousand words. However, I have yet to see the first master's thesis written in images alone. Textual information adds significant information to an image, and vice-versa. Much can be gained by linking images to text. When images and text occur together they most likely have something in common with one another. It would be useful to learn this relationship. Both modalities have their own characteristics, advantages and disadvantages. By combining them, parts of their disadvantages can be overcome.

With the advance of the digital revolution, the vast information on the Internet has come available to a large public. Keyword search engines like Google[1] are successfully applied to search through billions of web pages. However, multimedia documents are not easily described by keywords alone. New techniques for navigating this space are required, combining information found in all modalities.

Adding meaning to digital images is a hard problem. First, the sensory gap denotes the difference between the real object in the world and the computational representation of this object. Moreover, the semantic gap is the disparity between what one can extract from the visual data and the meaning a human being can distil from an image [32]. Complementing an image with words provides significant semantics. In image search engines based on the query-by-example strategy it is hard to find an initial image to give to the system as an example to find similar images. By adding text to images, the user can type some words and find relevant images. Adding visual information to words will also help to distinguish specific visual features. For example searching for a red bike using merely keywords will only succeed when such an image is sufficiently annotated with '`red bike`'. Using keywords and the image content, in this case its color, will substantially narrow the search.

Several practical applications for linking images and text may be identified [1]. Numerous organizations manage collections of images for internal use. Automatic image annotation can help such organizations to search and manage their collections. If people are not familiar with an image collection, they may wish to browse it. Browsing is a natural way for people to get acquainted with an image collection. In order to make this manageable the collection can be organized in a way that makes sense. Collecting images that look similar and are similarly annotated would provide a good start for organizing images.

[1] `http://www.google.com/`

1.1 Problem Statement

The main objective of this thesis is to create an information retrieval system that links visual features of an image and words of the corresponding caption together, in order to take advantage of the semantics added by combining both modalities. See figure 1.1 for a constructed example of the additional semantics gained by combining words and images.

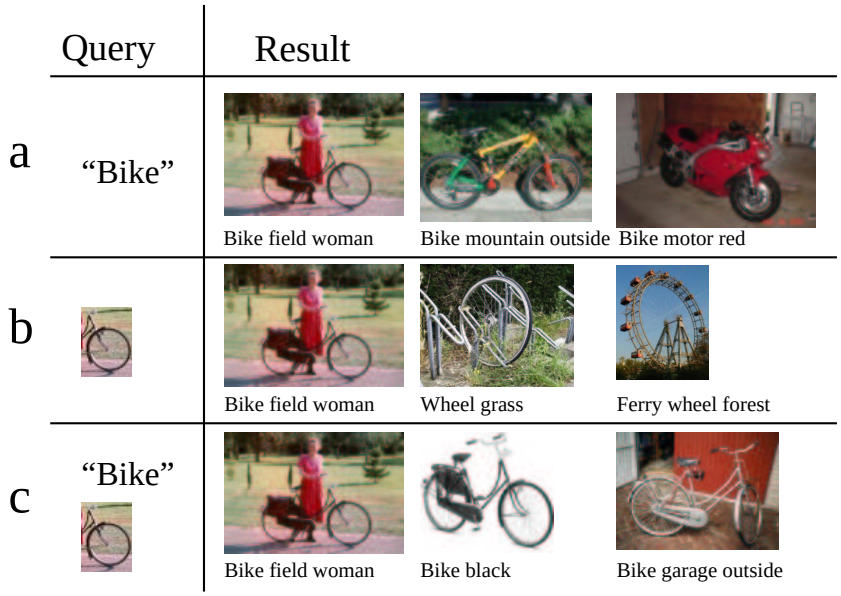


Figure 1.1: Constructed example of querying an image database. (a) using text only, (b) using images only, (c) combining images and text.

Words are made up of single letters and are therefore discrete in nature. An image is represented by pixels, however it's a continuous function. Therefore we propose to discretize the local image structure by approximating most of the local neighborhood of a pixel. These image patches represent the vocabulary of the images. Because an entire image is described by a single caption it is hard to learn which parts of an image belong to which words, if any direct relation exists at all. When images and words frequently occur together, there most likely is a relation between them. A learning algorithm can be used to cluster the words and the image patches. Latent Semantic Indexing (LSI) learns the most significant linear relationships between words and documents. Where a document in our case contains words and image patches. LSI maps frequently co-occurring terms and image patches on a concept. Ideally, this creates a single concept for term synonyms and image patches representing this concept.

1.2 Organization

In the next chapter we will introduce some history and background of information retrieval. Text retrieval and content based image retrieval are reviewed. Moreover, the textual model and the model used to learn relationships between image and text is presented in detail. Chapter 3 will describe the new method for image retrieval and the integration of words in an image with the

words in a text. Chapter 4 presents an evaluation of the image retrieval scheme and some examples of text retrieval, image retrieval and the combination of text and image retrieval. Chapter 5 will end the thesis by presenting the conclusions and the discussion.

Chapter 2

Information Retrieval

This chapter will introduce the main concepts of information retrieval, in particular text retrieval and content based image retrieval. Some history and background of information retrieval is presented, popular text retrieval schemes are described and a small review of content based image retrieval is given. The textual model as used in this thesis and the model used to learn relationships between image and text is presented in detail.

2.1 Introduction to Information Retrieval

What is information? The answer to this question seems imperative to build a system which main function is to retrieve it. An easy answer is not available. Many disciplines of science tried to conjure a definition. Shannon [30] describes a definition of information from a probabilistic viewpoint. Bateman [2] gives a more philosophical definition. Shannon claims that information is that which reduces uncertainty and Bateman states that information is that which changes us. Both definitions are very subjective. Information depends on the user. The first use of the term 'information retrieval' was in 1951 by Calvin N. Mooers. He intended the term to describe the conversion of a request for information into a useful set of references. He devised the term during the completion of his MIT master's thesis [24] in 1947 on zatocoding, an information retrieval system using punch cards. Everyone who has used a search-engine on the Internet is familiar with an information retrieval system. A schematic view of an information retrieval system is shown in figure 2.1.

Information retrieval is not database querying. Databases work with highly structured information. The data model of a specific database determines the possible queries a user can ask. Usually the form of the query will have to follow the data model[1]. A database is used where exact matching is required. Information retrieval models use highly ambiguous queries and need some amount of fuzziness as the user defines the search terms.

The key notion of information retrieval is that relevance is defined in terms of similarity. This assumes that if a document is similar to a query it is relevant. Similarity can be defined in several

[1] One may build additional layers upon the database, like a GUI or a natural language interface, but these abstractions must be translatable to the data model.

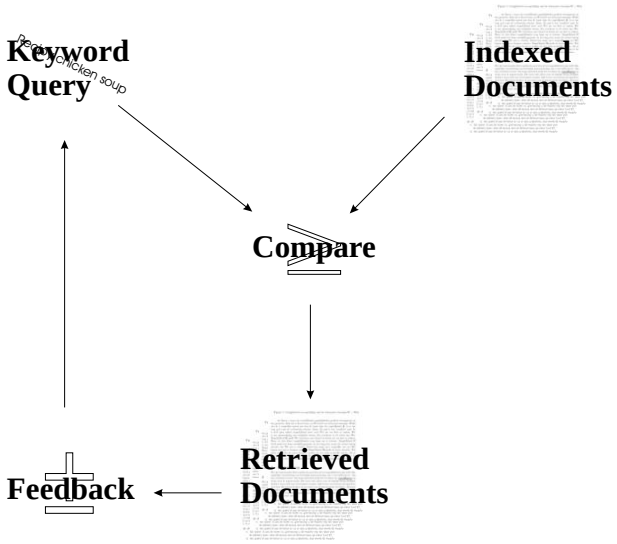


Figure 2.1: Schematic overview of an information retrieval system. The system has indexed a large set of documents. The user can query the system using keywords, whereupon the system retrieves the most relevant documents to this query. Some systems allow the user to provide feedback to the system about the relevancy of the retrieved documents. In this way, the system can take the preference of the user into account.

ways, depending on the modality of the information. For text retrieval, similarity is usually measured in the overlap of the words used in both query and document. For image retrieval, similarity is defined in how much an example image looks like an indexed image. The need to define relevance as similarity originates from the limited semantic value a computer can extract from a document. For text, the natural language processing community [12, 22] tries to find the meaning of a sentence using methods like part-of-speech tagging, parsing and semantic interpretation. Each sub-field has impressive results on limited domains, however, truly understanding generic sentences is an unsolved problem. Extracting the semantics for images is an even harder problem. For text in general, people tend to agree on the meaning of a given text. A language is a system of communication between people. Images can also be used in communication, think of sketches and technical drawings. However, a key difference between images and language is that the visual system is primary used to deal with the physical world, not for communication between people. For an image, different people see and notice different things. The information a system can extract from an image is even worse, as humans have substantial a-priori knowledge of the surrounding world. This discrepancy between human interpretation and extracted information is known as the semantic gap [32]. In addition to the semantic gap there is the lack of information inherent to using images of objects in stead of using the objects themselves. An image has a certain illumination, camera viewpoint, noise etc. This difference between real world objects and the information in a computational description derived from a recording of that scene is known as the sensory gap.

Information retrieval is historically linked to text retrieval. In the early days, the limited computer capacity restrained the processing of other sources of information. Examples of other media types are images, video, speech etc. We will proceed with a small overview of text retrieval methods, followed by an overview of image retrieval.

2.2 Text Retrieval

A document is relevant to a query if they are similar. Many approaches determine the similarity of a query to a document based on the words that are used. Most models use a bag-of-words approach. This means that the order between words is lost. It is surprising that this approach works successfully, as the following sentences[2] are considered equivalent:

Information retrieval, embraces the intellectual aspects of the description of information and its specification for search, and also whatever systems, techniques, or machines that are employed to carry out the operation.

Also and and are aspects carry description embraces employed for information information intellectual its machines of of operation or out retrieval search specification systems techniques that the the the to whatever

For human beings it is hard to distil the meaning of the second sentence. According to bag-of-words models they are equal.

Techniques from Natural Language Processing (NLP) [22] can be used to aid text retrieval. Accessing words in a document presumes that the input text is dividable in distinct words. This process is referred to as tokenization. The main cue of word separation in English is the occurrence of a space or a tab between words, however this is not necessarily reliable. Problematic words for tokenization are words with periods as in '`etc.`', with apostrophes like '`isn't`', with the use of hyphenation like '`bag-of-words`', with jargon as in '`Micro$oft`'. Words are subject to morphological changes, depending on their place and use in a sentence. The different representations can be stemmed to their basic form. Words like '`sit, sits, sat`' can be represented by '`sit`'. However, the use of stemming is debatable. As stemming is not perfect, incorrect documents can be retrieved. Examples of problematic word/stem pairs are '`police/policy, business/busy`'. Moreover, some terms are not suitable for stemming. A term like '`operating systems`' can be successfully stemmed, yet the stemmed form loses its meaning. Function words or stop words are only used to glue a sentence together and consequently dispensable in a bag-of-words model. Examples of stop words are '*a, the, therefore, who*'. Removal of stop words reduces the memory requirements and increases the computation speed. Stop lists are commonly available on the Internet.

A popular choice for text retrieval is the Boolean model. This model treats a document as a set of words and uses set-theoretical operators. A bag of words models words and their occurrence counts where a set of words removes count information. The terms in a query are linked together with the Boolean operators AND, OR and NOT. They are defined in the following way:

$$\begin{aligned} t_1 \textbf{ and } t_2 &= \{d|t_1 \in d\} \cap \{d|t_2 \in d\} \quad , \\ t_1 \textbf{ or } t_2 &= \{d|t_1 \in d\} \cup \{d|t_2 \in d\} \quad , \\ t_1 \textbf{ not } t_2 &= \{d|t_1 \in d\} \setminus \{d|t_2 \in d\} \quad , \end{aligned}$$

where d are documents, and t are terms. This model gives a user robust control over the retrieved documents. However, non-expert users find it hard to formulate the queries they want. Documents are, or are not, retrieved based on the occurrence of a word. A single wrong word in a document could rank a relevant document non-relevant. Moreover, the retrieved documents are all equally

[2] Quoted from Calvin Mooers.

ranked and the number of retrieved documents can vary from the whole collection to a single document.

A better alternative is given by the probabilistic retrieval model surveyed in [11]. The model tries to estimate the probability that a specific document d_m will be judged relevant to a specific query q_k

$$P(R) = P(R|q_k, d_m) \quad . \tag{2.1}$$

The model is based on the assumption that the terms are distributed differently in relevant and non-relevant documents. To model this assumption the notion of odds $O(R)$ is often used. Where

$$O(R) = \frac{P(R)}{1 - P(R)} \quad , \tag{2.2}$$

computes the ratio between the probability of a document being relevant and the probability that the document is not relevant. The bag-of-words notion is present in the assumption that the terms are probabilistic independent of each other. As this model uses the probability of relevance, adjusting the odds can incorporate user feedback by letting the user weigh the relevance of the retrieved documents.

A recent approach uses language models for retrieval, as surveyed by [18]. A language model tries to model the terminology used in a document. When language models are used for retrieval the probability that the language model of a document generates a query is computed. As unigram models (n-gram models with n=1) are used, this model follows the bag-of-words approach. N-grams are commonly used in natural language processing and model the probability of a word given the previous n words occurring before the word: $P(W_i|W_{i-1}, W_{i-2}, \ldots, W_{i-n})$. The probability that a query $(T_1, T_2, \ldots, T_N)$ of length n is sampled from a document D is defined by

$$P((T_1, T_2, \ldots, T_N|D) = \prod_{i=1}^{n}((1 - \lambda_i)P(T_i) + (1 - \lambda_i)P(T_i|D) \quad , \tag{2.3}$$

where $P(T)$ is the probability of drawing a term at random, $P(T|D)$ is the probability of drawing a random term from document D and λ_i is the user assigned relevance weight of the term.

2.3 Content Based Image Retrieval

In parallel to text retrieval, image retrieval considers an image similar to a query image if they look similar. This section concentrates on information retrieval aspects inherent to image retrieval. An overview paper by Smeulders et al. [32] giving a review of over 200 references in image retrieval is used as a guide. Low level image elements are defined in terms of color, shape and texture. The content of an image can be described by aggregating the low-level image properties to image features. The features can be used to compute similarity between images. Interaction and user feedback can be used to tune the retrieval system to the needs of the user.

Low-level image properties are distinguished in color, local shape and texture. Color lifts the one-dimensional gray world that we perceive at night to the three-dimensional spectral spectacle we see during the day. Color is a strong cue for object similarity. Two aspects of color are described: the variation of the recorded color and the human perception of color. Variations in the recording

can be caused by a different viewpoint, the position of the illumination source, the spectrum of the illuminant and the way the light interacts with the object. Humans have the ability to perceive the same apparent color with variations in the illumination, which change the physical spectrum of the perceived light. This ability is named color constancy.

Local shape denotes all properties that capture conspicuous geometric details in the image. For example edges, corners, curvature. The Gaussian scale space theory is the complete and unique primary step in preattentive vision, capturing all local information. Some elaboration on scale space theory is presented in section 3.2, p. 21.

Texture is defined as all what is left after color and local shape have been considered or it is defined by terms as structure and randomness. Typically, texture is composed of a repetitive seemingly random pattern of smaller elements, like the surface of a wall or the fabric of a cloth.

Invariance plays an important role in measuring low level image properties. The aim of invariant descriptions is to identify objects, no matter from how and where they are observed, at the loss of some of the information content. E.g. color invariants try to measure image content disregarding illumination differences. More information about color and rotational invariants is described in section 3.2, p. 25.

An image is often divided in parts. Four approaches are identified: strong segmentation, weak segmentation, signs and partitioning. Strong segmentation aims at a complete separation of the objects from the background. This is not likely to succeed for general domains. Weak segmentation groups image data in conspicuous regions each homogenous according to some criterion. Signs are objects with a near fixed shape, like an eye or a traffic light, only applicable in a narrow domain. A partitioning divides the image regardless of the data in predefined regions, e.g. 3x3 blocks.

Low level image properties are represented in features, describing the content of an image. Different feature representations are considered: global, salient or shape. Global features aggregate the spatial information of a partitioning, describing the entire image regardless of the image data. An alternative is given by using salient features. Salient features use weak segmentation, distilling the information in the image in a limited number of feature values. This reduction motivates the use of the most informative parts of the image. Shape features are used with strong segmentation describing the identified object. The brittleness of strong segmentation constrains this approach to narrow domains.

Similarity of feature values is a measure of image similarity. Computing the distance between features creates an ordering of all images based on some similarity measure. Some distance measures are reported. If F_q and F_d are histograms, their distance can be measured as the Mahalanobis distance $(F_q - F_d)^t \Sigma^{-1} (F_q - F_d)$, where Σ is the covariance matrix between F_q and F_d, the histogram intersection $\sum_{i=1}^{n} \bigcap_i F_d(i), F_q(i)$ and the Minkowski distance $\sum_{i=1}^{n} |F_q(i) - F_d(i)|$.

2.4 Vector Space Model

Text is modeled using a bag-of-words approach. Besides the loss of all sentence structure, each word is considered independent of all the other words. This bag-of-words approach does not reflect reality. The occurrence of the term '`retrieval`' in a document will give the word '`information`' a higher probability of co-occurring in the document than any random word. Moreover, synonyms

have exactly the same meaning but a complete different form. Words are not independent of each other. This motivates the modeling of a co-occurrence relationship. Latent Semantic Indexing (LSI) tries to model the underlying relations between words and between documents. In what follows we will describe the bag-of-words model used for text and the extension of this model to LSI.

The bag-of-words model used in this thesis for text, is the Vector Space Model [26]. The model is based on linear algebra. A document is modeled as a vector of words where each word is a dimension in Euclidean space. Let $T = \{t^1, t^2, \ldots, t^n\}$ denote the set of terms in the collection. Then we can represent the terms d_j^T in document d_j as a vector $\vec{x} = (x_1, x_2, \ldots, x_n)$ with:

$$x_i = \begin{cases} t_j^i & \text{if } t^i \in d_j^T \quad ; \\ 0 & \text{if } t^i \notin d_j^T \quad . \end{cases} \tag{2.4}$$

Where t_j^i represents the frequency of term t^i in document d_j. Combining all document vectors creates a term-document matrix. An example of such a matrix is shown in figure 2.2.

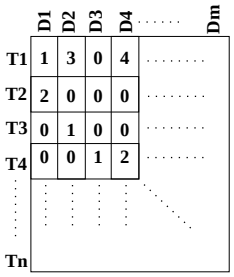


Figure 2.2: A Term Document matrix.

Similarity $\mathcal{S}$ of a query to a document is defined as the Euclidean distance of the query vector $\vec{q}$ to a document vector $\vec{d}$:

$$\mathcal{S} = \vec{q} \cdot \vec{d} \quad , \tag{2.5}$$

where $\cdot$ denotes the inner product.

The advantages of the vector space model include a ranked result of the retrieved documents, the possibility to enter free text and no strict matching of the documents. The ranked result is an ordering on the distance of a document to the query vector. Such an ordering will determine the similarity of a document to the query. Free text search eliminates the use of a difficult query language as in the Boolean model. The matching is not strict, in the sense that a query containing multiple words will also retrieve documents where not all words are present.

Depending on the context, a word has an amount of information. In an archive about information retrieval, the word '`retrieval`' does not add much information about this document, as the word '`retrieval`' is very likely to appear in many documents in the collection. A term can be given a weight, depending on its information content. A weighting scheme has two components: a global weight and a local weight. The global importance of a term is indicating its overall importance in the entire collection, weighting all occurrences of the term with the same value. Local term weighting measures the importance of the term in a document. The value for a term i in a document j is $L(i,j) * G(i)$, where $L(i,j)$ is the local weighting for term i in document j and

$G(i)$ is the global weight. Several different term weighting schemes have been developed. In [9] several methods are tested in combination with LSI. A method based on Shannon's entropy [30] performs best. The words are weighed in an information theoretical manner where words with high probability get a low weight. The rationale is that words that occur frequently in the complete collection have low information content. However, if a single document contains many occurrences of a word, the document is probably relevant. Taken both considerations into account, the weight w_j^i of a term i in a document j is given by

$$w_j^i = \quad L(i,j) * G(i) = \quad \left[\log(t_j^i + 1)\right] * \left[1 - \sum_j^{ndoc} \frac{P(t_j^i)\log(P(t_j^i))}{\log(ndoc)}\right] \quad , \tag{2.6}$$

where t_j^i is the number of times term t^i occurs in document d_j, $ndoc$ is the number of documents in the collection and $P(t_j^i)$ is the probability of t_j^i, $P(t_j^i) = t_j^i / f^i$, where f^i denotes the total number of times word t^i occurs in the collection. The reason for incrementing the number of terms in document d_j with one in the local weight, is to avoid $log(0) = \infty$. The logarithms dampen the effect of very high term frequencies. By subtracting the entropy in the global weight from a constant, a minimum weight is assigned to terms that are equally distributed over documents and maximum weight to terms that are concentrated in a few documents.

Large documents containing much more words are more likely to be closer to a query vector. They have a higher probability to contain a query word because they are made up of many words. As this is unfair for smaller documents, the document and query vectors are scaled to unity length. This lowers the weights of the terms in large documents. Dividing the document vector by its length performs scaling:

$$\tilde{d}_m = \frac{d_m}{|d_m|} \quad . \tag{2.7}$$

When dealing with vectors of size 1, the cosine of the angle between $\vec{q}$ and $\vec{d}$ is equivalent to the Euclidean distance, scaling the similarity between -1 and 1:

$$\mathcal{S} = Cos(\vec{q}, \vec{d}) \quad . \tag{2.8}$$

2.5 Latent Semantic Indexing

Latent Semantic Indexing (LSI) [8] models the underlying, latent, relation between documents and words. A problem with the Vector Space Model is that the words used in a query are often not the same as the words used in a relevant document. Similar words can have multiple meanings and different words can have the same meaning. These phenomena are named polysemy and synonymy, respectively. For example the words `car` and `automobile` are synonyms where the polysemous word `jaguar` can either mean a large cat or can denote a type of car. The goal of LSI is to find a model of the true relationships between terms and documents.

In order to achieve this goal, a truncated Singular Value Decomposition (SVD) is applied on the term-document matrix. In accordance with the original paper [8] we do not stem words to their basic form. We use a stop list of 571 common words and omit terms occurring in only one

document, as they contribute little or nothing to the SVD solution. The truncated SVD projects the high dimensional term-document matrix to a matrix with less dimensions. This reduced matrix is the closest matrix in the reduced space, in a least-square sense, to the original matrix. An example of dimension reduction from two dimensions to 1 dimension is displayed in figure 2.3.

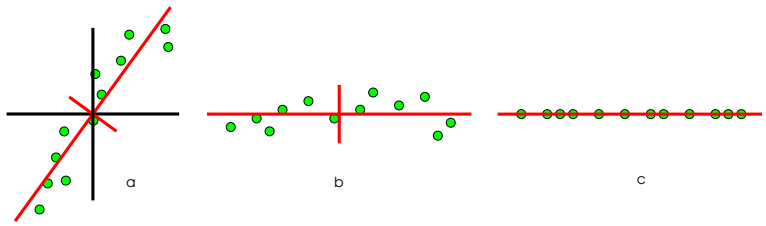


Figure 2.3: An example of dimension reduction, from 2 dimensions to 1 dimension. (a) Two-dimensional datapoints. (b) Rotating the data so the vectors responsible for the most variation are perpendicular. (c) Reducing 2 dimensions to 1 dimension so that the dimension responsible for the most variation is preserved.

Singular Value Decomposition is a generalization of Factor Analysis. In Factor Analysis a square n dimensional matrix is decomposed in 2 matrices containing the eigenvectors and the eigenvalues of the original matrix. The eigenvectors are all perpendicular to each other and span the high dimensional space described by the original matrix. Each eigenvector is accompanied with an eigenvalue. This value denotes the size of the eigenvector. This size corresponds with the variation of the dimension of the eigenvector found in the original matrix.

Definition 2.5.1 (eigenvector and eigenvalue) *If $\mathcal{A}$ is a $n \times n$ matrix, then a nonzero vector x in $\mathcal{R}^n$ is called an **eigenvector** of $\mathcal{A}$ if x is a scalar multiple of x; that is,*

$$\mathcal{A}x = \lambda x \quad , \tag{2.9}$$

*for some scalar λ. The scalar λ is called an **eigenvalue** of $\mathcal{A}$, and x is said to be an eigenvector of $\mathcal{A}$ corresponding to λ.*

Singular Value Decomposition extends Factor Analysis to deal with non-square, arbitrary rectangular matrices. The $m \times n$ dimensional matrix is decomposed in 3 matrices. Two matrices contain the singular vectors and one diagonal matrix the singular values.

Definition 2.5.2 (Singular Value Decomposition) *The Singular Value Decomposition of an $m \times n$ matrix of full rank $\mathcal{A}$ (with $n > m$) is given by*

$$\mathcal{A} = \mathcal{T}\mathcal{S}\mathcal{D}^t \quad , \tag{2.10}$$

where $\mathcal{T}$ is an orthonormal $m \times m$ matrix, $\mathcal{D}^t$ is an orthonormal $n \times n$ matrix and $\mathcal{S}$ is a $m \times n$ diagonal matrix. $\mathcal{S} = diag(\sigma_1, \sigma_2, \ldots, \sigma_n)$ ordered so, that $\sigma_1 > \sigma_2 > \ldots \sigma_n$

The generalization of Factor Analysis is based on the fact that any matrix multiplied with its transposed is a square matrix. When the matrix $\mathcal{A}$ is not of full rank, SVD will reduce dependent

vectors to zero as they are a linear combination of other vectors. The matrix $\mathcal{T}$ contains the eigenvectors of the square matrix $\mathcal{A}\mathcal{A}^t$ and $\mathcal{D}$ contains the eigenvectors of the square matrix $\mathcal{A}^t\mathcal{A}$. In both cases, the diagonal matrix $\mathcal{S}^2$ contains the eigenvalues.

SVD allows a simple strategy for an optimal approximate fit using smaller matrices. If the singular values in $\mathcal{S}$ are ordered by size, the first k largest values may be kept and the $(n-k)$ lower values set to zero. This gives an approximate diagonal matrix $\tilde{\mathcal{S}}$. The product $\mathcal{T}\tilde{\mathcal{S}}\mathcal{D}^t$ of this matrices is a matrix $\tilde{\mathcal{A}}$ which is only approximately equal to $\mathcal{A}$, and is of rank k. It can be shown that the new matrix $\tilde{\mathcal{A}}$ is the matrix of rank k, which is closest in the least squares sense to $\mathcal{A}$. This yields the new reduced model:

$$\mathcal{A} \approx \tilde{\mathcal{A}} = \mathcal{T}\tilde{\mathcal{S}}\mathcal{D}^t \quad . \tag{2.11}$$

Setting values to zero in the diagonal matrix $\mathcal{S}$ will set several rows and columns of the matrices $\mathcal{T}$ and $\mathcal{D}$ to zero. The model can be simplified by deleting the rows and columns of all the matrices, which will be set to zero. Matrix $\mathcal{S}$ will become a $k \times k$ diagonal matrix, $\mathcal{T}$ will become a $m \times k$ matrix and $\mathcal{D}$ will reduce to a $k \times n$ matrix.

The choice of the value k is critical for good results. Ideally, we want a value of k that is large enough to fit all the real structure in the data, but small enough so that we do not overfit the data. Displaying the values in $\mathcal{S}$ gives some idea where a cut-off could take place. The lower the values, the less variation is present in the data. A more theoretical sound approach can be taken using the minimum description length (MDL) method [17]. The MDL approach deals with the dimension selection problem by taken into account both the length of the model and the length of the description of the data using this model. A generic model will be small but needs much space to describe the data. An overfitting model on the other hand will reduce the description of the data, but this will use a large model. The optimal solution will be a small model that still accurately describes the data. Hence, using the smallest combination of model and description of the data with the model will give a good value for dimensional reduction.

How do we compare a query to the documents in the reduced space? We start with its term vector $\mathcal{X}_q$ in the non-reduced space and transform it to a pseudo-document $\mathcal{D}_q$ in the reduced space that we can use just like a row of $\mathcal{D}$. In a non-reduced space, an existing document $\mathcal{X}_i$ would map to its corresponding row $\mathcal{D}_i$. By rewriting the equations we can arrive at a formula for computing $\mathcal{D}$ when $\mathcal{X}$, $\mathcal{T}$ and $\mathcal{S}$ are given. As the derivation is not given in the original paper we will give it here:

$$\begin{aligned}
X &= TSD^t && \text{(definition SVD)} \\
X &= T(DS^t)^t && (A^tB^t = (BA)^t, \text{ rewrite D}^\text{t} \text{ to D}) \\
X &= T(DS)^t && (S^t = S, \text{ S is a diagonalmatrix}) \\
X &= (DST^t)^t && (A^tB^t = (BA)^t) \\
X^t &= DST^t && (\text{rewrote D}^\text{t} \text{ to D}) \\
X^tT &= DS && (TT^t = I, \text{as T is orthonormal}) \\
X^tTS^{-1} &= D && (S^{-1}S = I) \\
D &= X^tTS^{-1} && (\text{Change sides})
\end{aligned}$$

This last formula to calculate a given query vector $\mathcal{X}_q$ to a row $\mathcal{D}_q$ in $\mathcal{D}$ is also valid for the reduced space, which gives the formula:

$$\mathcal{D}_q = \mathcal{X}_q^t \mathcal{T} \tilde{\mathcal{S}}^{-1} \quad . \tag{2.12}$$

Determining the similarity of documents i and j is done by calculating the dot product of rows i and j of matrix $\mathcal{DS}$. As described in the previous section, the matrix $\mathcal{D}$ contains the eigenvectors of the square matrix $\mathcal{A}^t\mathcal{A}$ which contains the dot product values of the document vectors with the document vectors. The matrix $\mathcal{D}$ is scaled to unit length by the SVD. To scale the matrix D to proportion it has to be multiplied with the corresponding singular values found in $\mathcal{S}$. Comparing document i with document j in matrix $\mathcal{D}$ is just taking the dot product of row i with row j in the matrix $\mathcal{DS}$. In this way, a query can be mapped in the low dimensional space and compared to indexed documents.

It is also possible to calculate the r most relevant terms to the query. When a query vector is projected in the reduced space, it is possible to construct a new row in the approximate matrix $\tilde{\mathcal{A}}$ which corresponds with this query where the matrix $\tilde{\mathcal{A}}$ is the new (term $\times$ document) matrix approximating the original matrix $\mathcal{A}$. This query vector in the reduced space contains a value for each term that denotes its relative importance, compared to the other terms. When the first r most relevant terms are required, the values have to be sorted, and the first r term vectors will represent the most relevant terms. Computing a new row $\tilde{\mathcal{A}}_q$ from $\tilde{\mathcal{A}}$, corresponding to a query q follows by using $\mathcal{D}_q^t$ rather than $\mathcal{D}$ in formula 2.11:

$$\tilde{\mathcal{A}}_q = \mathcal{T} \tilde{\mathcal{S}} \mathcal{D}_q^t \quad . \tag{2.13}$$

Providing the user with the most relevant terms adds extra information compared to only retrieving the documents. The user may see related words to use in subsequent queries and gets information about what the system 'thinks' is important. The latter gives a user more understanding why the system retrieves certain documents.

This chapter provides the methodology to learn relations between entities. The vector space model treats a document as a bag of words where the words are independent. LSI is used to learn the relation between the documents and words. The next chapter will decompose an image in words, making the models as introduced for text applicable on images.

Chapter 3

Visual Model

This chapter introduces a new image retrieval scheme, based on Latent Semantic Indexing applied on invariant local color structure. The same bag-of-words approach as described in chapter 2 is used, where the order of the words does not matter.

Global features are very popular in content based image retrieval. In contrast with global features, local descriptors are very suitable both for image retrieval as for the equivalent of a word in an image. Global features are highly sensitive to noise. Object occlusion and image additions distort the global shape, texture or color information of the image. When part of the image is not visible due to occlusion the global description of the image lacks this information. Moreover, in a global model of an image the object and the background information are merged, in which case the object representation depends of the background. Some examples of the problems encountered in image retrieval based on global features are shown in figure 3.1.

a b c d

Figure 3.1: Examples of problems with image retrieval with global features. (a) The original image, (b) adding information to the image (c) occluding the object (d) both adding information to the image as occluding the object.

The problem can be solved by grouping part of the data, hence segmenting the image into regions [32]. However, a strong segmentation of an arbitrary image is not likely to succeed. Segmentation is unreliable and results in incoherent fragmented regions. Poor segmentation can result in inconsistent regions for further similarity matching. We believe that for a generic image retrieval scheme, local features are required.

Words bind their underlying characters to a single meaningful entity. The same is required for images. Not a single pixel should be used, rather a small patch of pixels, capturing the local structure of the image. Using only pixel values in a bag-of-words or histogram approach will

remove too much structure from the image, nevertheless Swain and Ballard [35] use this approach successfully in object recognition. By using only pixel color values the images a, b and c in figure 3.2 are considered equivalent. Clearly image b has all spatial structure removed. When using pixel blocks, images a and c in figure 3.2 are similar. There is still a lot of variation between both images but compared to image b they are much more alike. The overlap at the boundaries between the patches introduces spatial correlation between the blocks. This correlation makes it possible to reconstruct the original image in fig 3.2a from the jigsaw puzzle in fig 3.2b.

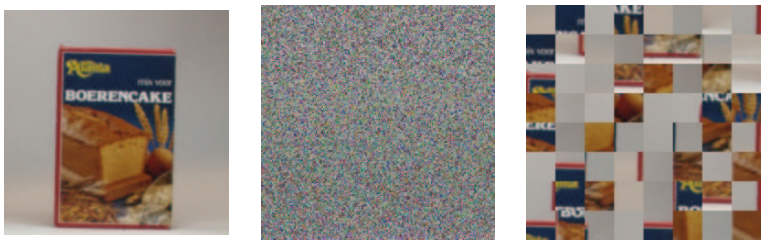


Figure 3.2: Three equivalent images (a) The original image, (b) the same image using a bag-of-words approach with pixel values (c) the same image using a bag-of-words approach with blocks of pixels.

3.1 Literature Overview

Several papers in the literature propose local features that may be used as the 'words' in an image. A popular method is to tessellate the image in square blocks of pixels, and then describe each block using color, texture or shape features. In [21] such a tessellation is used in the construction of visual keywords. They define a visual keyword as a manually constructed set of pixel blocks each set containing similar semantic concepts like `face`, `crowd`, `building`. Each keyword has a vector of image features consisting of color features in YIQ space and Gabor filters for texture information. Visual keywords are used to label each block of an image using a fuzzy membership algorithm. They aim at adding meaning to images by using semantic concepts in contrast to low level image features. In [33] text retrieval techniques like inverted files, term weighing and relevance feedback are applied on tessellated blocks of the image. Inverted files make high dimensional feature spaces accessible, while term weighing and relevance feedback are used to determine important blocks. The features used are HSV color and Gabor texture filters. Zhu et al. [38] propose keyblock-based image retrieval. They denote a keyblock as the representative of an unsupervised clustering of pixel blocks. Each pixel blocks in an image is then labeled with their corresponding keyblock. Local spatial correlation between keyblocks is modeled with a n-gram structure (see section 3, p. 10). Using tessellated blocks of pixels has some drawbacks, as these methods use no color invariance nor 2D rotational invariance. Translational invariance is limited to the case where an entire block has changed position. Identical objects photographed upside down and under different light intensity will yield different representations.

A different approach by Schmid et al. [28] uses local gray value invariant salient points for image retrieval. Salient points are local features with high informational content, in their case points where the signal changes in two directions. For these salient points, the third order local 2D

rotational invariant jet is computed at several scales. Their method deals with problems as partial matching and occluded images. A problem with this approach is misdetection of the salient points, which are not guaranteed to be the same after a rotation of the image. Moreover, the salient points are typically found at the edges and corners of the object. When the object is placed in a different background, as for example figure 3.1 (b), the corners and edges are exactly the points where the most change occurs. Hence, the algorithm may fail.

The misdetection of salient points is circumvented by taking histograms of local multi scale gray value structure information for every pixel in the image [27]. Translation invariance is given by the use of histograms and 2D rotational invariance is achieved by generating several rotated versions of the original image in steps of 45^o. The technique proves robust for small-scale differences and 3D rotated images. Histograms of local features are also used by Siggelkow et al.[31]. They use explicit invariant features for rotation and translation, eliminating the need to generate rotated versions of the original image.

Not so many authors deal with the integration of text and image. We present an overview of relevant work.

A method dealing with the relationship between an image and its caption in a narrow domain is presented in [34]. The Piction system is described, a system that identifies human faces in newspaper photographs based on the information contained in the associated caption. The limited domain makes sophisticated natural language processing possible in order to extract spatial information about the image. The approach nicely illustrates the advantages of integrating text and images.

Relevance feedback is utilized in [37] to weigh relations in a semantic network. Relations between texts, visual features and their combination is modeled. Several predominant global features are used to describe an image. By using relevance feedback the user can indicate the importance of both an image and of keywords. Relations between relevant keywords and relevant images are learned. The method is very flexible, making it possible to learn relations solely based on user preference. However, no automatic extraction of implicit semantic relations in the data is performed.

Tessellated blocks of an image are used in [25] to relate images and words. Each block is described with the color histogram and 45^o rotated Sobel filters for measuring gradient intensity. The blocks are reduced to prototypes by using vector quantization. As each block inherits all words of the annotation of the complete image, the prototypes have accumulated all words of the clustered blocks they stand for. This aggregation of words is used to calculate the most probable words for a prototype. Experiments indicate an increase over random word assignment of 9%.

A machine learning technique called Multiple-Instance learning is used in [23] to classify images of natural scenes based on their keyword categorization. Multiple-Instance learning is a technique to discriminate in a set of positive and negative labeled bags. An image is considered a bag of small blocks and the keyword label determines the class of the image. Experiments show the technique capable to retrieve images with similar classification. The problem with this approach is that it can only handle single classes, assigning only one keyword to an image.

Several probabilistic learning models used for matching image and text are evaluated in a paper by Barnard et al. [1]. They describe several versions of linear and hierarchical clustering methods [19] as used for images and words [10], and Latent Dirichilet Allocation (LDA) for modeling annotated

images [5]. They define *annotation* as the prediction of text given an image. Both methods are extended to a machine translation approach trying to model regions with text. Naming each region with words is denoted *correspondence*. All models use segmented images to relate image regions to words, where words occurring less than 20 times are removed. Both annotation and correspondence is evaluated. Annotation is tested by $(r/n - w/(N-n))$ where w is the number of words predicted right, w denotes the words predicted wrong, N is the vocabulary size and n is the actual number of words for the image. Correspondence is measured by hand, as no dataset with a ground truth is available. Tests are conducted on images in the training set, held out images, and on substantial different held out images. For annotation, hierarchical models and linear models perform similar. LDA does not perform well on the training set, however the results are competitive with the other models when the test data are novel. Correspondence results are somewhat better then annotation results. The authors conclude that words are generally associated with pieces of images, not with the entire image. Future work lies in the integration of a thesaurus, supervised learning by user feedback and a sound evaluation of the correspondence relation.

Text retrieval and image retrieval is combined in [7]. This method spiders the World Wide Web for images and supplements the images with weighed words found in the HTML file. Two different vectors are created for an image, one vector for image features and one vector for text features. The image features consist of the color histogram [35] and the dominant orientation histogram [29]. The text vector is the low dimensional vector of the words after applying LSI on the text collection. The two vectors are concatenated to combine images and text, thus excluding image features from the LSI.

Benitez and Chang [3] use several clustering techniques on image features and text. The words are stemmed, filtered for stop words and words appearing less than 5 times are removed. Two versions of term weighting are used: tf·idf and the same log·entropy as we use (see section 2.4, p.12). After term weighting, the LSI is applied on the text only. The image features include the color histogram, texture and edge histograms. Experiments were conducted with a pre-labeled collection of images and their textual annotation. As a measure of evaluation they compute an entropy-based score, indicating the quality of the clusters with respect to the labels. The cluster algorithms used are: k-means, Ward algorithm, k-Nearest Neighbors, Self-Organizing Map and Linear Vector Quantization. Experiments were performed using text only, image only, and image features concatenated with text. Only the best results are presented, yielding the best results for text-only clustering, then, the image features with concatenated LSI-text, followed by images-only. The authors conclude that both text and image features can be used to cluster meaningfully, as both approaches do better than random. Moreover, they conclude that visual features and text descriptors are highly uncorrelated, as LSI does not make a significant difference. Therefore the authors believe text and image should be integrated.

In [16] LSI is used on a small dataset of 50 images. Image retrieval experiments are conducted with the color histogram and tessellated blocks. The result shows an increase in performance when using LSI. Further experiments include supplemented keywords to the images, with a total of 15 different keywords, where a keyword denotes the classification of the image. As a keyword reveals the class of the image, results improve when using keywords. As the experimental dataset is very small, the presented results show no significant difference between all methods and tend to lie close together.

LSI is performed on the image content combined with the textual context in [36]. Feature descrip-

tions of tessellated blocks of the image include the HSV color space and Gabor texture filters. The author notes the difference between the distribution of the visual features and the distribution of the textual terms. Two approaches are adopted to match both distributions: an equally sparse set of image terms by attaining a large range of possible feature values and an equally small vocabulary size of the image terms by limiting the possible feature values. Experiments are conducted for text-only, images-only and the text and image combination. For both approaches the retrieval results of the combined text and image index greatly resembles the images-only experiments with an overlap of more than 80%. Visual similarity dominates the model. Nevertheless, examples indicate retrieval performance of image and text combined outperforms both text and image based approaches.

We believe histograms of local features are similar to a bag-of-words approach. A 'word' in an image describes the local behavior of the image at a point. We propose to extend the local feature histogram method to color and use well defined techniques from scale space theory to describe the local structure. Color is an important cue for object similarity and several color invariance techniques can be used to achieve a robust method for lighting and shading differences. Moreover, the proposed method models explicit 2D rotational invariant local structure, removing the need to generate rotated versions of the image. Using Latent Semantic Indexing on the 'words' will learn relationships between frequently co-occurring image patches. Moreover, our new image feature model captures a more suitable equivalent of a word in an image. Hence, making an improvement possible for the relation between word and image.

This chapter is organized as follows. First, the notion of local spatial structure for gray value images is described. Next, the Taylor series is explained, which provides the basis for both the extension to the color model, as for the modeling of the local image structure. Then the Gaussian color model is introduced, followed by the description of the discritization of the continue values that make up local image structure. The chapter is concluded with the integration of image words with textual words.

3.2 Multi Scale Local Color Jet

This section will introduce the basic mathematical tools to model the local color structure of an image. We will first describe the representation and approximation of local structure in a gray value image and then extend these methods to the color domain.

An image is a representation of a physical observation. This can be expressed mathematically as a function of the electromagnetic energy E over a spatial domain D:

$$f : D \mapsto E \quad . \tag{3.1}$$

A two dimensional gray value image $f : \mathbb{R}^2 \mapsto D$ has a value $f(a, b)$ at each location (a,b).

The local structure of a function can be described with the Taylor series, the polynomial approximation of any differentiable function in a point. The quality of the approximation around a point increases with the degree of the polynomial. The n-th degree Taylor polynomial approximating a 1-dimensional function $f(x)$ in point a is given by

$$f(x+a) = f(a) + f'(a)x + \frac{f''(a)}{2!}x^2 + \cdots + \frac{f^{(n)}(a)}{n!}x^n + O(n+1) \quad . \tag{3.2}$$

Figure 3.3 shows an example of such a Taylor approximation of the function e^x around the point $x = 2$.

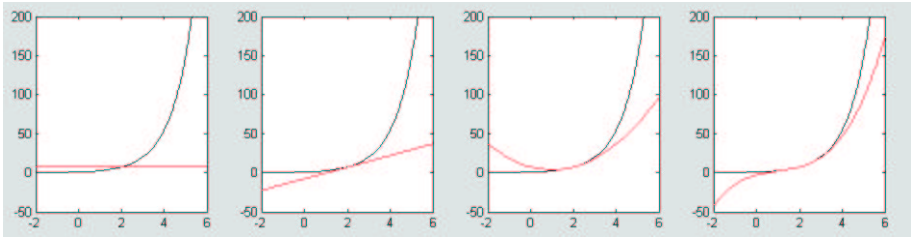


Figure 3.3: Example of a Taylor approximation at $x = 2$ of the function e^x. The red line is the approximation, the black line is the original function e^x

When extending the Taylor series to two dimensions we can approximate an image function. The two-dimensional Taylor series is created by supplementing the polynomial for the x coördinates with an additional polynomial for the y coördinates. Approximating a two-dimensional function $f(x, y)$ around the point a gives:

$$\begin{aligned} f(a_x + x, a_y + y) \quad \approx \quad & f(a_x, a_y) + x\partial_x f(a_x, a_y) + y\partial_y f(a_x, a_y) + \\ & \frac{1}{2!x^2}\partial_{xx} f(a_x, a_y) + \frac{1}{2!y^2}\partial_{yy} f(a_x, a_y) + 2\frac{1}{2!xy}\partial_{xy} f(a_x, a_y) + \cdots \end{aligned} \tag{3.3}$$

Image information is highly dependent of scale. The mathematical model assumes a continues function, however image information is limited by its resolution. High-resolution images contain details of the scene while low-resolution images only store crude image information. The Taylor series in equation 3.3 uses the partial derivatives of a function. The partial derivative in the x-direction is defined by

$$\partial_x f(a, b) = \lim_{\gamma \to 0} \frac{f(a + \gamma x, b) - f(a, b)}{\gamma} \quad . \tag{3.4}$$

However, because images are typically stored as discrete pixel values with given resolution, γ can not approximate 0. It is necessary to create a 'proper' differentiable function from f. The discrete image has to be smoothed. The only way to do this, under general constraints, is by using a convolution with a Gaussian function [20]. A convolution of function f with function g is defined as:

$$(f \otimes g)(x, y) = \int \int f(x - i, y - j)g(i, j) \, \mathrm{d}i \, \mathrm{d}j \quad . \tag{3.5}$$

An example of the 1-dimensional and 2-dimensional Gaussian function can be seen in figure 3.4. The 1 dimensional Gaussian kernel is given by

$$G_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} \, e^{-\frac{x^2}{2\sigma^2}} \quad . \tag{3.6}$$

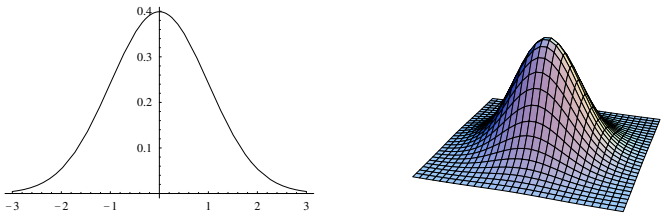


Figure 3.4: The 1-dimensional and 2-dimensional Gaussian function, with $\sigma = 1$.

The size of the Gaussian kernel depends on the value of σ. This value determines the amount of smoothing of the image and thus the scale of observation. The effect of convolving with different values of σ can be seen in figure 3.5. When keeping σ a free parameter, a multi scale structure emerges. This notion of scale is inherent to our perception of the world. Given an image of several trees, there is no reason to assume a-priori interest in the forest, a tree or in the leaves. Scale of interest has to be explicitly modeled in the description of the local structure.

Figure 3.5: Using gaussian convolution at scale $\sigma \in \{1, 2, 4\}$ to smooth an image.

The derivatives of an image are accessible via the linear properties of the convolution. Taking the derivative of the function $(f \otimes g)$ is equal to convolving function f with the derivative of g:

$$(\partial f) \otimes g = \partial(f \otimes g) = f \otimes (\partial g) \quad . \tag{3.7}$$

This means that convolving with the derivative of the Gaussian yields the derivative of the Gaussian convolved image. Using this property, a whole family of higher order derivatives comes available by convolving the image with higher order derivatives of the Gaussian.

Up to now, the techniques all deal with gray, scalar images, i.e. image functions with a single value at each location. This domain needs to be extended to color. Color provides much extra information. Some edges are only available in the color domain and color can help determine object color similarity, indistinguishable in gray images. Figure 3.6 shows an arbitrary spectrum of light that may fall on the camera or the human eye.

Color is defined in terms of human observation. There is no one-to-one mapping of the spectrum of a light source to the perceived color. The human color processing apparatus down-samples the infinite dimensional color space to three dimensions, yielding similar colors with different spectral distributions. The color model described by [13] fits very well into the Gaussian model for treating scalar images. The manner of describing the structure of an image is extended in a similar way to the wavelength domain. Furthermore, a range of invariant properties are available for this

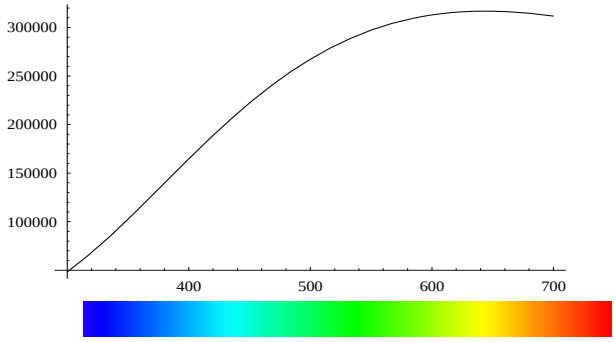


Figure 3.6: An arbitrary spectrum (the energy E distribution over the wavelength λ.

model [14]. In analog to gray value images, the derivatives of a smooth spectral distribution are taken. These can be used to approximate the spectral distribution with a Taylor series. If $\mathrm{E}(\lambda)$ is the energy distribution of the light, where λ describes wavelength, then the Taylor approximation at a central wavelength λ_0 is

$$E(\lambda) \approx E(\lambda_0) + \partial_\lambda E(\lambda_0)\lambda + \frac{\partial_{\lambda\lambda} E(\lambda_0)}{2}\lambda^2 + \cdots \tag{3.8}$$

Similar to the spatial domain, the derivatives of the energy distribution E come available with the linear properties of the convolution. As this involves a Taylor approximation in a single point, the convolution reduces to an integration with a Gaussian $G(\lambda_0, \sigma_\lambda)$ at the single point λ_0 with a scale of σ_λ where

$$\begin{aligned} E(\lambda_0) &= \int E(\lambda) G(\lambda, \lambda_0, \sigma_0)\,\mathrm{d}\lambda \\ \partial_\lambda E(\lambda_0) &= \int E(\lambda) \partial_\lambda G(\lambda, \lambda_0, \sigma_0)\,\mathrm{d}\lambda \\ \partial_{\lambda\lambda} E(\lambda_0) &= \int E(\lambda) \partial_{\lambda\lambda} G(\lambda, \lambda_0, \sigma_0)\,\mathrm{d}\lambda \end{aligned} \tag{3.9}$$

These derivatives are plotted in figure 3.7.

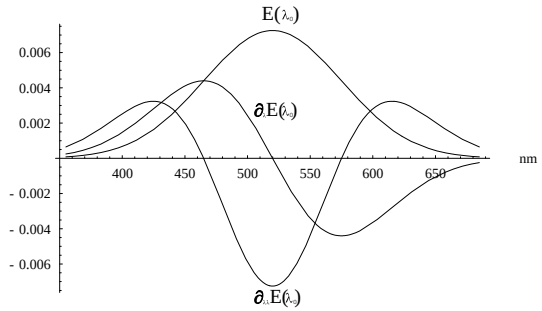


Figure 3.7: The zeroth, first and second derivative of the Gaussian function with respect to the wavelength.

The human visual system uses three dimensions, yielding a second order approximation of the

spectral information. In accordance with the human visual system, the Gaussian color model uses second order spectral information. The zeroth order derivative measures the luminance, the first order derivative the 'blue-yellowness', and the second order the 'red-greenness' of a spectrum. The Gaussian color model uses the first three components of the Taylor expansion of the Gaussian weighted spectral distribution, however a RGB image is measured in the Red Green and Blue sensitivity components of the light. The RGB sensitivities have to be transformed to the Gaussian spectral derivatives. In [13] an optimal transformation matrix with the Taylor expansion in the point $\lambda_0 = 520$nm and with a Gaussian spectral scale of $\sigma_\lambda = 55$nm is derived under the assumption of standard REC 709 CIE RGB sensitivities:

$$\begin{bmatrix} E \\ \partial_\lambda E \\ \partial_{\lambda\lambda} E \end{bmatrix} = \begin{pmatrix} 0.06 & 0.63 & 0.27 \\ 0.3 & 0.04 & -0.35 \\ 0.34 & -0.6 & 0.17 \end{pmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.10}$$

The spectral and spatial derivatives can be used in a Taylor approximation of the local color structure of an image. The set of N Gaussian derivative kernels for the three color channels, including the zeroth order derivative is referred to as the color N-jet. An example of a Taylor approximation in a single pixel using the color N-jet is shown in figure 3.8. The figure demonstrates that local color and spatial information can be obtained from the derivatives. Note that only the middle of the image can be well approximated. As the Gaussian used in the convolution falls off to zero at the tail of the distribution the Taylor approximation becomes unstable at points further away from the origin.

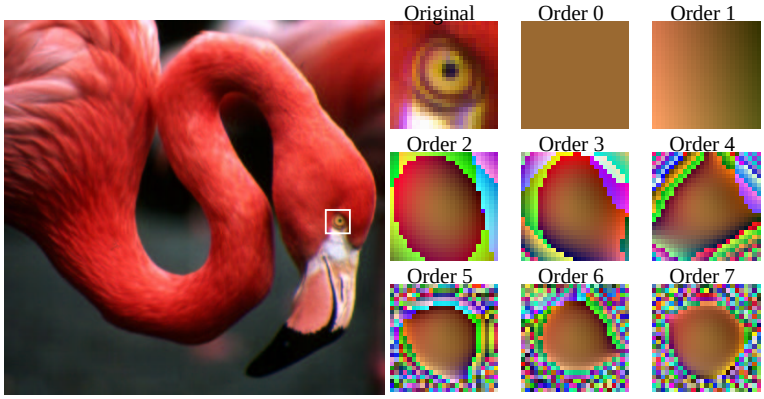


Figure 3.8: Approximating local color structure with a Taylor expansion in a single pixel, using $\sigma = 3$.

When comparing images of the same object, differences in measurement due to the scene environment pose a problem. Taking two pictures of an object yields two different representations of the same scene. Differences in lighting conditions and in camera rotation change the recorded measurements of the scene. Image invariants deal with the problem to measure the information in a scene independent of properties not inherent to the recorded object. Color invariance aims at keeping the measurements constant under different intensity, viewpoint and shading. In [14] several of these color invariants are described. We will test two invariants for our method. We use uniform intensity invariance, which gets rid of uniform luminance differences between images. An

invariant property $\mathcal{W}$ is described for uniform local illumination

$$\mathcal{W}_{\lambda^m x^n} = \frac{\partial_{\lambda^m x^n} E}{E} \qquad m \geq 0, n \geq 1 \quad , \tag{3.11}$$

where E is the energy. The rationale is, that the reflected light of an object depends on the intensity of the light which illuminates the object. Thus normalizing for intensity creates measurements invariant of illumination. We also test for uneven intensity invariance, which removes reflectance independent of the viewpoint, surface orientation, illumination direction and illumination intensity. An invariant property $\mathcal{C}$ is described as

$$\mathcal{C}_{\lambda^m x^n} = \partial_{x^n} \frac{\partial_{\lambda^m} E}{E} \qquad m \geq 1, n \geq 0 \quad , \tag{3.12}$$

where E is the energy. The invariant may be interpreted as the spatial derivative of the intensity normalized spectral slope $\partial_\lambda \mathcal{C}$ and curvature $\partial_{\lambda\lambda}\mathcal{C}$. As this property is only valid for $\partial_{\lambda^m} E$ where $m \geq 1$, the invariant cannot be used for gray valued images. The difference between invariants $\mathcal{W}$ and $\mathcal{C}$ is in the assumption of illumination. In $\mathcal{W}$ the complete spatial and spectral derivatives are normalized for luminance. This models the assumption that the local spatial neighborhood as described by the Taylor series of the spatial structure is illuminated with the same energy E. The shadow invariant $\mathcal{C}$ normalizes the spectral information with the energy E and computes the spatial derivatives independent of the spectral energy. This makes the local spatial neighborhood invariant for intensity changes like shadow.

Rotational invariance aims at keeping values constant when rotating the image under the camera, for example the images in figure 4.1. The process of finding the local N-jet for gray images is described in section 3.2, p.21. These derivatives are taken in the (x,y)-coordinates of the camera. When the coördinates change the values of the derivatives also change, while the local image structure remains constant. In order to deal with this variation, we rotate the derivatives of each pixel to the direction where there is the most change. Hence, rotating all the derivatives to the direction of the gradient, as the gradient does not change when the angle of the camera is rotated. Rotating derivatives to the direction of the gradient yields the following formula for the n^{th} derivative in the x-direction and the m^{th} derivative in the y-direction:

$$\partial_{v^n}\partial_{w^m} = \big(\sin\theta\frac{\partial}{\partial x} \; - \; \cos\theta\frac{\partial}{\partial y}\big)^m \big(\cos\theta\frac{\partial}{\partial x} \; + \; \sin\theta\frac{\partial}{\partial y}\big)^n \qquad n, m \geq 0 \quad , \tag{3.13}$$

where θ is the angle of the gradient, $\tan\theta = \frac{\partial y}{\partial x}$. For color images we chose to rotate each channel to the direction of the gradient in the luminance, the zeroth order derivative with respect to the wavelength λ. This rotates the spectral derivatives with the same angle θ, and makes equation 3.13 applicable to each color channel.

In summary, the local spatial and color structure in an image can be represented by a truncated Taylor series at a pixel. By using the Gaussian color model the color information fits naturally in this framework. Moreover, image invariants measure intrinsic properties of the scene, independent of luminance and camera rotation.

3.3 Representing Local Color Structure

The continues values of the color N-jet lack the natural discrete nature of words in a text. The color N-jet needs to be discretized. Labeling each element of this discrete partition of the color N-jet creates words representing the local color structure of an image. Each word then says something about the behavior of the image at a certain pixel. Similar image patches are described with the same word. Local structure with similar color and similar curvature are then grouped. Discretizing the local color N-jet yields a vocabulary that can be used to describe local spatial color structure.

A multidimensional histogram is used for the discretization process. The dimensions of the histogram are equal to the number of derivatives in the color N-jet. Each spatial spectral derivative is partitioned in equally sized bins each with their own bin label. The bin labels in this context represent a single character and the concatenation of all these characters make up a word. As this word is determined by the color N-jet, it models the local spatial color information. For example a first order spatial Taylor approximation with a second order spectral approximation of a pixel E_0 yields words with a length of 6 letters. This because the multidimensional histogram uses these 6 values:

$$(E_0, \partial_x E_0, \partial_y E_0, \partial_\lambda E_0, \partial_{\lambda x} E_0, \partial_{\lambda y} E_0, \partial_{\lambda\lambda} E_0, \partial_{\lambda\lambda x} E_0, \partial_{\lambda\lambda y} E_0) \quad .$$

Figure 3.9 visualizes the creation of the words with first order structure for a gray value image. Each pixel of the image is labeled in this manner.

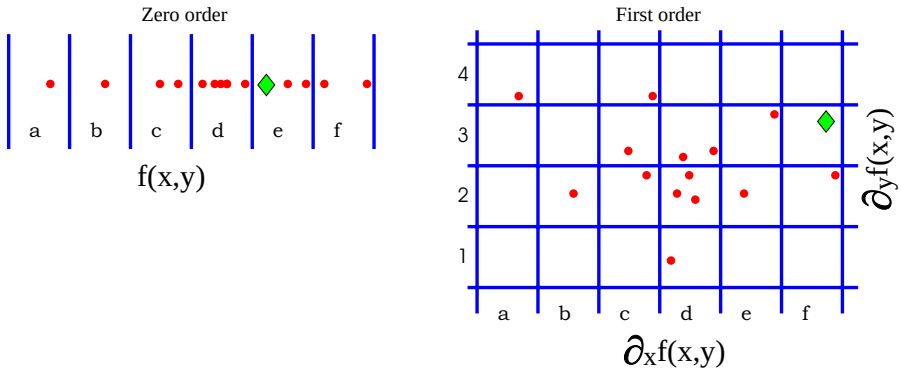


Figure 3.9: The labeling process for zeroth order derivatives (left) and for the first-order derivatives (right). The green diamond on the left and right represent the derivatives of one pixel. This pixel is labelled as EF3.

The partitioning of the color N-jet is in a way related to image segmentation. For, both methods group similar image structure. Image segmentation aims at separating object from background. Hence grouping object pixels and background pixels. The problem with image segmentation is that the notion of 'object' and 'background' are only defined in a semantic interpretation of the image. What is, or is not an object that depends on the context, meaning and use of the image. To an engineer, the segmented image of a ship will consist of the welding plates, the propulsion system, the pipelines running along the deck, e.g. In contrast, a tourist will give a completely different interpretation of the segments of the ship yielding e.g. swimming pool, sun deck and dance floor.

The labeling of the color N-jet makes no such effort. It groups low level image features to describe the local behavior of the image and operates at a low semantic level.

The grouping of similar structure can be visualized. If we represent each word with a different color, identical colors in a single image represent identical words. Using this visualization, we can display the effects of applying the invariants discussed in the previous section. Figure 3.10 shows the effect on the vocabulary of an image when using different invariants.

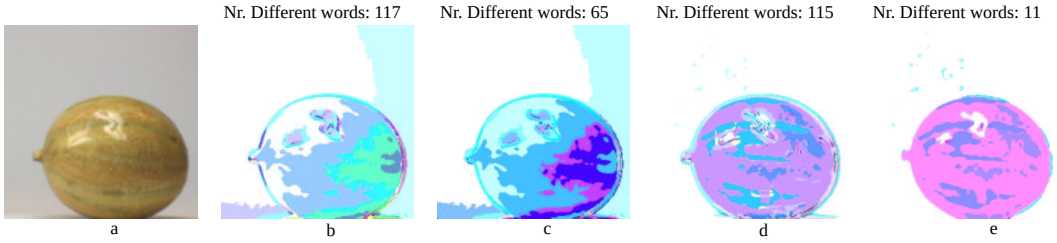


Figure 3.10: The effect of using invariance on the vocabulary. Each word is represented with a different color. Identical colors in a single image represent identical words. The size of the vocabulary is displayed above each labeled image. The labeling is done with the second order color jet using 17 bins for each value at a scale of $\sigma = 2$ pixels. (a) The original image. (b) Labeling the image using no invariance. (c) Labeling the image with rotational invariance. Note the contour of the object is labeled everywhere with the same word and the vocabulary size decreases. (d) Labeling the image with rotational and luminance invariance $\mathcal{W}$. Note the disappearance of intensity changes. The vocabulary size increases, as shadows are labeled incorrectly. (e) Labeling the image with rotational and shadow invariance $\mathcal{C}$. Note the significant reduction of the words in the vocabulary and the disappearance of the shadow of the object.

Latent Semantic Indexing (LSI) (see section 2, p.7) maps frequently occurring terms close to each other in a reduced space, thus learning relationships between frequently co-occurring words. In the image domain, the words describe local color structure. Using LSI on these words thus learns relationships between frequently co-occurring local image structures. LSI will retrieve images beyond the information present in the image query, finding images containing local structures frequently co-occurring with the local structure in the query. In the text domain this increases the number of retrieved relevant documents as synonyms are found. A similar increase in recall should be expected of the use of LSI on the image domain.

Moreover, the text retrieval techniques term weighing and vector normalization (see chapter 2.4, p.12) are applied on the images. The term weighing scheme will automatically give a high weight to patches occurring frequently in a single image and appearing non-frequently in the total collection. In this manner, the more informative patches are given higher weight in determining similarity than the less informative patches. Vector normalization makes the method robust for dealing with different image sizes. We use the cosine distance to measure similarity between vectors, (see chapter 2.4, p.13), as is often done in text retrieval.

3.4 Combining Image and Text

Combining text and images in our model is a matter of concatenation the textual word vector to the image feature vector. As each pixel in the image is labeled with a word, the visual information dominates the model. This problem has been identified by [36]. Their solution is restricting the visual model to match the sparseness or vocabulary size of the textual model. We believe a restriction on the model severely limits the information in the visual representation. Therefore, we deal with the model imbalance by normalizing the visual and textual information to equal importance. Similar to vector normalization (see section 2.4,p. 13), we can scale the length of the text vector to the same length of the image feature vector. If the combined model is given by $\vec{tv}$ and vector $\vec{t}$ represents the textual model and $\vec{v}$ denotes the visual model the normalization is given by

$$\vec{tv} = \vec{t}\,\frac{\mid \vec{v} \mid}{\mid \vec{t} \mid} :: \vec{v} \quad , \tag{3.14}$$

where :: is the concatenation operator. This normalization is also applied on the query. Using LSI on this combined space learns co-occurrence relations between: words and other words, image patches and words, image patches and other image patches.

In summary, we define the words in an image as a set of labels where each pixel is assigned a label describing the behavior of the image in the pixel. A label stands for a discretized signature of the color N-jet, making up the local spatial and spectral structure of the image in the pixel. Text and image information can easily be combined in the same semantic space by concatenating the textual word vector and the visual word vector. Performing LSI on the combined space will relate text to text, image to image and text to image. The next chapter will evaluate the image retrieval scheme, including some examples of the combined search on images and text.

Chapter 4

Experiments

Latent Semantic Indexing for text retrieval has proven more effective then standard Vector Space retrieval [8, 9]. Our new visual model based on LSI has yet to demonstrate its quality in content based image retrieval. This chapter will measure the effectiveness of the new visual retrieval model. A dataset of 430 images of common household articles is indexed and a queryset of 100 different images of the same objects photographed under different conditions is used to query the model. Furthermore, this chapter will demonstrate the advantages of using the combination of images and text for retrieval. As no suitable standard dataset is available to measure the retrieval performance of combining image and text, we leave it at several examples.

The practical issues of doing experiments require an implementation, parameter settings and pragmatic handling of the theory. We implemented LSI in C++ using svdpackc [4] for a fast Singular Value Decomposition. The SMART stoplist[1] of 571 words is used to remove stop words. We use no stemming. As LSI learns the relationships between frequently co-occurring entities, we remove all terms that occur only in one single document, to save time and memory. For image processing we use the Horus[2] library of the ISIS group at the University of Amsterdam. When performing experiments several parameters can be considered. We have chosen to calculate the N-jet up to order 2 as this adds curvature information and still is feasible to index. The range for assigning bins or characters to the N-jet values of each pixel is set between -1 and 1. If a value is out of bounds, it is assigned to it's closest bin.

The intensity of the N-jet deteriorates for derivatives taken at a large scale. We therefore scale the N-jet according to:

$$\tilde{\partial}_{x^n} = \sigma^n \partial_{x^n} \qquad \tilde{\partial}_{y^m} = \sigma^m \partial_{y^m} \tag{4.1}$$

Small scale Gaussian convolution of $\sigma = 2$ is used to utilize prototypical, small image patches at narrow scale.

Color invariant indexing becomes unstable at low intensity, since color values are normalized by the intensity, and dividing by zero yields infinity (see chapter 3.2, p.25). Therefore, if the intensity is below 5% of the intensity range, we use the original non-invariant values of the N-jet.

[1] `ftp://ftp.cs.cornell.edu/pub/smart/english.stop`
[2] `http://www.science.uva.nl/~horus/`

4.1 Evaluating the Visual Model

In order to evaluate different aspects of our retrieval method we performed several experiments with a database of 430 images containing camera rotated images, occluded images and 2D rotated images. We experimented with grayscale and color indexing, both with and without intensity invariance and rotational invariance. The experiments are conducted for three different image query sets also used in [15]. The dataset $D_1, \ldots, D_n$ contains 430 images. The largest queryset is the 2D rotated objects with 70 query images. Figure 4.1 contains an example of the images in the dataset with corresponding query images. We will refer to this set as the '`standard`' queryset. Figure 4.2 contains an example of another image query set varying in camera angle, the images

Figure 4.1: Example images of the '`standard`' queryset. The images in the dataset (left) with their corresponding query images (right).

in the dataset are shown with corresponding query images. The camera angle s is changed for $s \in \{45, 60, 75, 80, 85\}$. This set has 4 different objects with a total of 20 images. We will refer to this set as the '`camera angle`' queryset.

Figure 4.2: Example images of the '`camera angle`' queryset. The image in the dataset (left) with it's corresponding query images varying in camera angle $s \in \{45, 60, 75, 80, 85\}$ (right).

Figure 4.3 contains an example of the images varying in occlusion percentage. The images in the dataset are shown with corresponding query images. The percentage of occlusion o varies for $o \in \{50, 65, 80, 90\}$. This set has 4 different objects with a total of 16 images. We will refer to this set as the '`occlusion`' queryset.

Figure 4.3: Example images of the '`occlusion`' queryset. The image in the dataset (left) with it's corresponding query images varying in the percentage of occlusion $o \in \{50, 65, 80, 90\}$ (right).

For a measure of recognition quality, let rank r^{Q_i} denote the position of the correct match for

query image Q_i in the ordered list of the D_n results of a query. This ranking gives $r^{Q_i} = 1$ for a perfect match and $r^{Q_i} = D_n$ when the required image is retrieved last. Ranking all the queries Q_m for their corresponding image in the dataset and scaling between 0 and 100 gives the average ranking principle:

$$\bar{r} = (\frac{1}{Q_m} \sum_{j=1}^{Q_m} \frac{D_n - r^{Q_i}}{D_n - 1})100\% \tag{4.2}$$

The color histogram is a popular method used for retrieval. A color histogram is comparable to our method when using only the zeroth order derivative in the labeling. This uses only the color values of the smoothed image, discarding local structure. Color histograms are known to be robust for occlusion and invariant for 2d rotation. To establish a baseline we test LSI on color histograms, using only the color pixel values. In the original paper [35] the authors advocate a small number of bins, grouping similar color. We used 64 bins for each color channel. We vary the size of the LSI dimension keeping it a parameter in the results. The higher the dimensions of the reduced space, the more accurate the approximation of the original space becomes mapping less co-occurring terms together. The highest dimension is most similar to the original color histogram.

Figure 4.4: Results for using LSI on color histograms using 64 bins for each color channel. The vocabulary size or the number of bins used is shown in the legend.

The retrieval results for color histogram indexing are displayed in figure 4.4. With color histograms LSI already performs better at a lower dimension than at the highest dimension. This already justifies the use of LSI for image retrieval. Further experiments will evaluate our new method, with local color structure rather than single pixel values. We will now present the results for second order spatial structure. The experiments use 19 bins for each element in the N-jet. We performed experiments for both grayscale and for color. Grayscale uses only the values for intensity, e.g. the smoothed spectral channel E. The LSI dimension is kept as a free parameter. For color indexing we used the color N-jet. The retrieval results for plain gray values and color indexing are displayed in figure 4.5. The retrieval results for rotational invariance are shown in figure 4.6. The retrieval results for the $\mathcal{W}$ and $\mathcal{C}$ color invariants, both with rotational invariance are shown in figure 4.7.

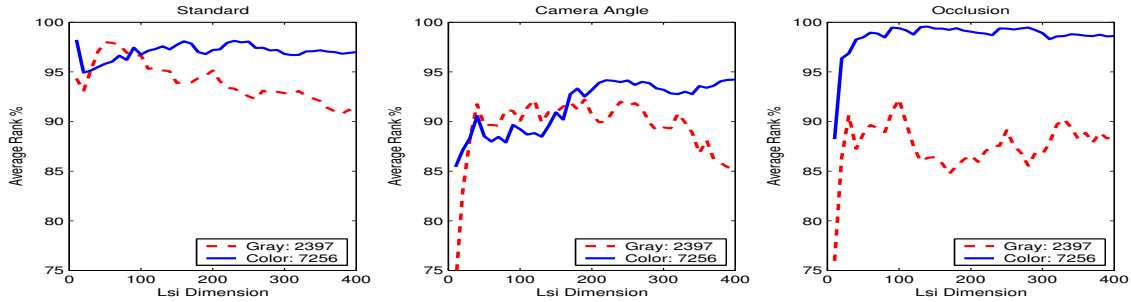


Figure 4.5: Results for color and gray indexing. The histograms use 19 bins. The vocabulary size is shown in the legend.

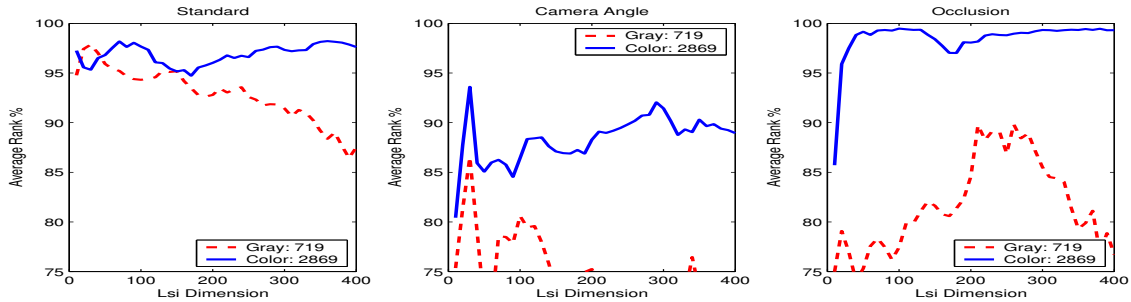


Figure 4.6: Results for rotation indexing. The histograms use 19 bins. The vocabulary size is shown in the legend.

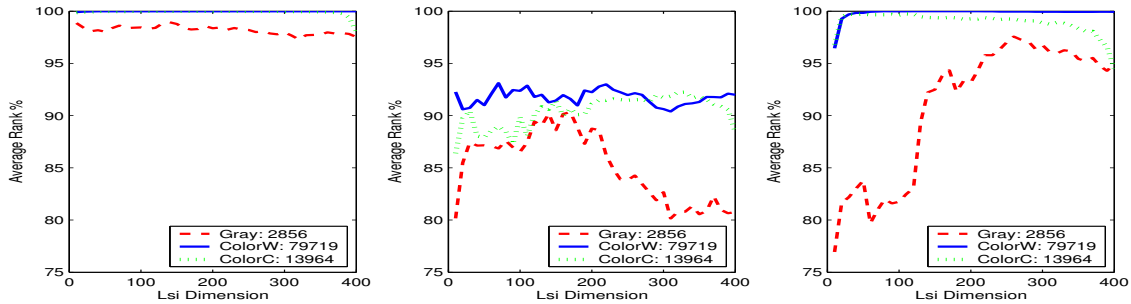


Figure 4.7: Results for the two color invariants with rotational invariance. The $\mathcal{C}$ color invariant cannot be used with grayscale images, as all intensity is discarded. The histograms use 19 bins. The vocabulary size for all methods is shown in the legend.

All methods perform better at a lower dimension, warranting dimension reduction. The best overall results are achieved with color indexing, using both rotational invariance and color invariance $\mathcal{W}$. Rotational invariance performs equally, or slightly worse than using no invariance at all. Color invariant $\mathcal{W}$ in general outperforms the $\mathcal{C}$ invariant. The lack of performance of rotational invariace and the $\mathcal{C}$ color invariant can be explained by the loss of information when using invariance, illustrated by the significant reduction in the vocabulary size of these invariants. The $\mathcal{W}$ color

invariant does not suffer from this problem, as already demonstrated in figure 3.10,p.28. Color presents an important cue for object similarity. Only when no color invariance is used, grayvalue indexing sometimes outperforms color indexing, indicating the increase in performance when using invariants.

Comparing the results of figures 4.5, 4.6 and 4.7 with figure 4.4 show an increase in performance for the '`standard`' dataset.

The vocabulary size is a measure of the complexity and the memory requirements of the method. The vocabulary represents the power of the words to express the image. In concurrence with text, a large vocabulary of words makes more elaborate word usage possible, while a small vocabulary limits the distinctiveness. The size of the vocabulary is displayed in the graphs of the retrieval results. Note the differences in vocabulary size between color and gray value indexing and between different invariants. The power of invariant descriptors has already been illustrated in figure 3.10, p. 28. The number of bins is directly related to the vocabulary size. A large number of bins, can distinguish between small shape and color differences, where a small number of bins labels small differences with the same word. As can be seen by the vocabulary size in figures 4.5, 4.6 and 4.7 invariant descriptors label similar structure with the same word, yielding a smaller vocabulary size with an equal number of bins. We believe that similar vocabulary size is a better measure than a similar number of histogram bins to compare performance differences between methods.

We will now present experimental results giving the same expressive power to each model. The histogram binsize is adjusted in a way that the size of the vocabulary used to describe the images is kept approximately constant. As the color N-jet uses approximately 3 times the information used for grayscale indexing, the latter can take advantage of a smaller binsize to arrive at a predefined constant vocabulary size. Grayscale indexing can thus use a finer sampling of the truncated N-jet yet lacking any color information. A very high vocabulary size becomes problematic for the memory requirements of our implementation of the SVD. We use a constant vocabulary size of about 75.000 labels. In the retrieval results the LSI dimension is used as a free parameter which is varied between 10 and 400. The results for plain gray values and color indexing are displayed in figure 4.8. The retrieval results for rotational invariance are shown in figure 4.9. The retrieval results for the $\mathcal{W}$ and $\mathcal{C}$ color invariants, both with rotational invariance are shown in figure 4.10.

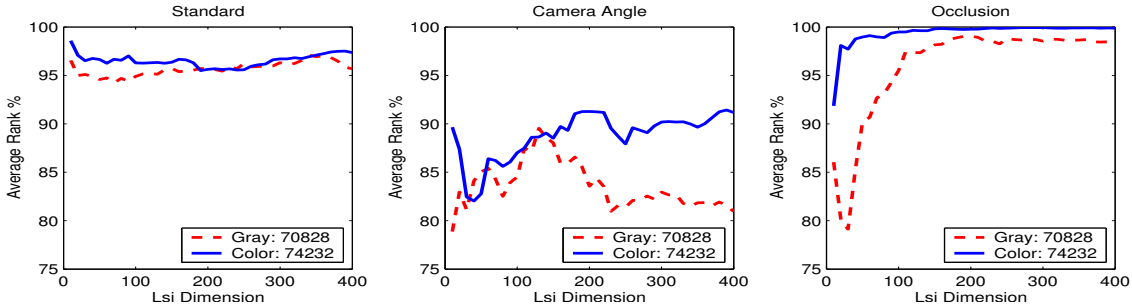


Figure 4.8: Results for color and gray indexing. The histogram for grayscale indexing uses 51 bins. Color indexing uses 33 bins. The vocabulary size is shown in the legend.

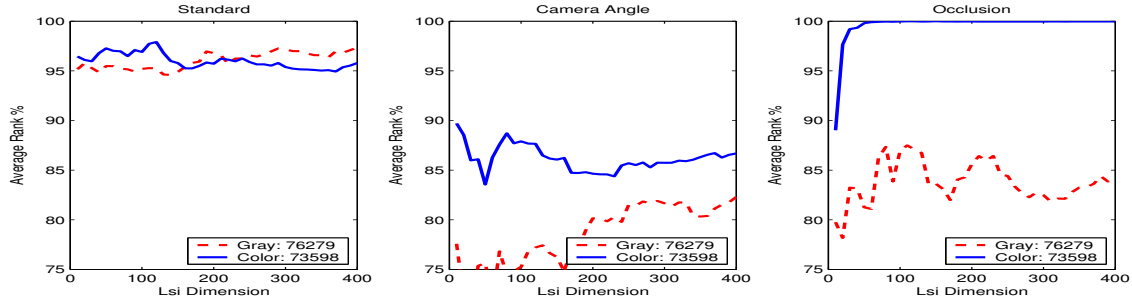


Figure 4.9: Results for rotation indexing. The histogram for grayscale indexing uses 83 bins. Color indexing uses 41 bins. The vocabulary size is shown in the legend.

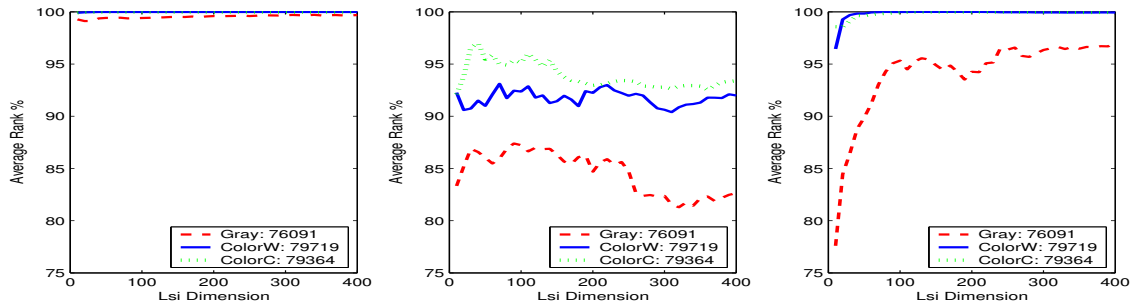


Figure 4.10: Results for the two color invariants with rotational invariance. The $\mathcal{C}$ color invariant cannot be used with grayscale images, as all intensity is discarded. The histogram for grayscale indexing with the $\mathcal{W}$ invariant uses 69 bins. Color indexing with the $\mathcal{W}$ invariant uses 19 bins and color indexing with the $\mathcal{C}$ invariant uses 37 bins. The vocabulary size for all methods is shown in the legend.

The best overall results are achieved with color indexing, using both rotational invariance and color invariant $\mathcal{C}$. The loss of information by using invariants is compensated by expressive power, yielding color invariant $\mathcal{C}$ more powerful. However, the added expressive power by adapting the binsize to create equal vocabulary sizes can not compensate for using rotational invariant indexing

only. Furthermore, the dataset seems to be exhausted by our method, as we perform near 100% for both the '`standard`' and the '`occlusion`' test collections. The '`camera angle`' score peaks at 97%.

Compared to [15] we perform somewhat equally well, e.g. near 100% with the '`standard`' test collection. In [15] the results for '`occlusion`' and '`camera angle`' are plotted against occlusion percentage and camera rotation. To compare our results, we kept the LSI dimension fixed at our best average dimension 40 and plotted the results for camera angle and rotation in figure 4.11.

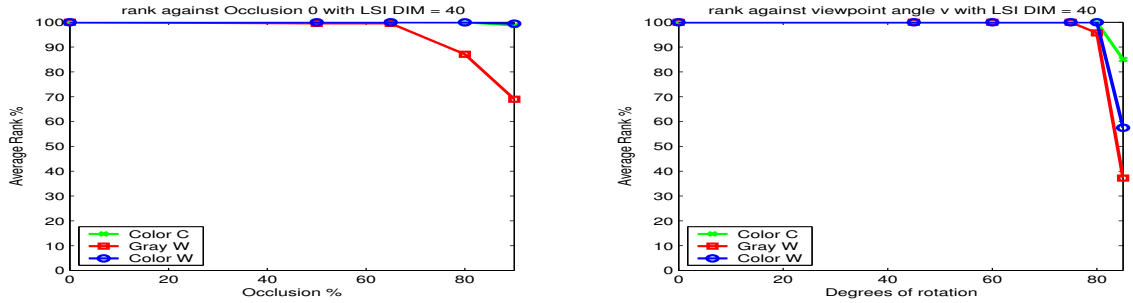


Figure 4.11: Results for color and gray intensity invariant indexing compared to the percentage of occlusion (left) and the angle of 3d camera rotation (right).

We score a lot better both for '`occlusion`' and for the variation in '`camera angle`'[3]. This confirms the power of the local features. These features are comparable to the bag-of-words-approach used for text. The next section will give some examples of combining images and text for informatino retrieval in a single model.

4.2 Text and Imaging Results

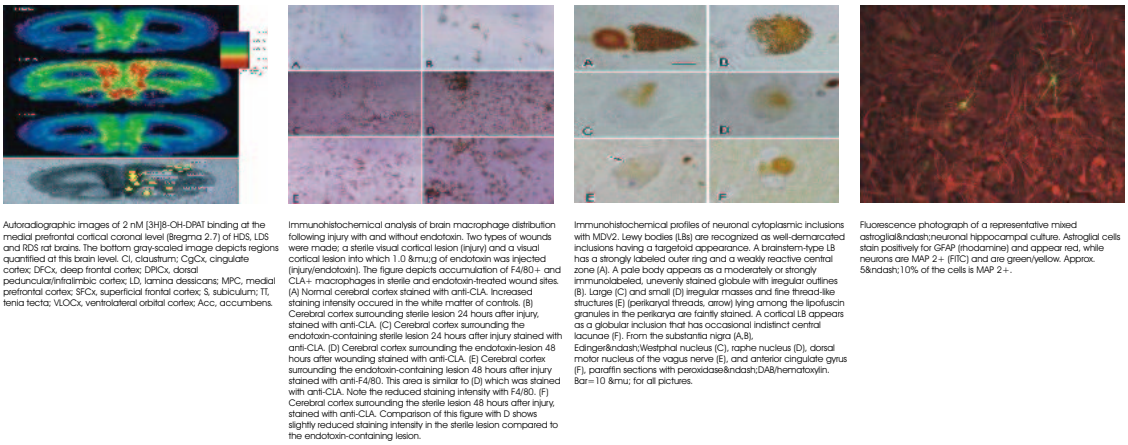
The combination of images and text is tested on the Elsevier data set. This set is comprised of biomedical images with a corresponding textual caption. There are nearly 745 color images in this set from scientific publications on biomedical research. The captions vary in size and description. See figure 4.12 for an excerpt of the dataset.

This section gives some examples of queries and their results. We will try and recreate the constructed example in figure 1.1, p.4. All examples are conducted with the best performing setting, using both rotational invariance and color invariant $\mathcal{C}$. Moreover, LSI provides the tools to calculate the most relevant terms to a query. As described in section 2.5, p.15, the relevant terms to a specific query can be computed with

$$\mathcal{D}_q = \mathcal{X}_q^t \mathcal{T} \tilde{\mathcal{S}}^{-1} \quad . \tag{4.3}$$

Applying equation 4.3 on the image domain, enables us to calculate the most relevant local structures to an image query. This can give a user significant more information about the reason why an image is retrieved. We will display some queries with the 10 most relevant terms to the query,

[3]however in [15] the results for 85 degrees rotation are not included



Autoradiographic images of 2 nM [3H]8-OH-DPAT binding at the medial prefrontal cortical coronal level (Bregma 2.7) of HDS, LDS and RDS rat brains. The bottom gray-scaled image depicts regions quantified at this brain level. Cl, claustrum; CgCx, cingulate cortex; DFCx, deep frontal cortex; DPICx, dorsal peduncular/infralimbic cortex; LD, lamina dessicans; MPC, medial prefrontal cortex; SFCx, superficial frontal cortex; S, subiculum; TT, tenia tecta; VLOCx, ventrolateral orbital cortex; Acc, accumbens.

Immunohistochemical analysis of brain macrophage distribution following injury with and without endotoxin. Two types of wounds were made; a sterile visual cortical lesion (injury) and a visual cortical lesion into which 1.0 μg of endotoxin was injected (injury/endotoxin). The figure depicts accumulation of F4/80+ and CLA+ macrophages in sterile and endotoxin-treated wound sites. (A) Normal cerebral cortex stained with anti-CLA. Increased staining intensity occured in the white matter of controls. (B) Cerebral cortex surrounding sterile lesion 24 hours after injury, stained with anti-CLA. (C) Cerebral cortex surrounding the endotoxin-containing sterile lesion 24 hours after injury stained with anti-CLA. (D) Cerebral cortex surrounding the endotoxin-lesion 48 hours after wounding stained with anti-CLA. (E) Cerebral cortex surrounding the endotoxin-containing lesion 48 hours after injury stained with anti-F4/80. This area is similar to (D) which was stained with anti-CLA. Note the reduced staining intensity with F4/80. (F) Cerebral cortex surrounding the sterile lesion 48 hours after injury, stained with anti-CLA. Comparison of this figure with D shows slightly reduced staining intensity in the sterile lesion compared to the endotoxin-containing lesion.

Immunohistochemical profiles of neuronal cytoplasmic inclusions with MDV2. Lewy bodies (LBs) are recognized as well-demarcated inclusions having a targetoid appearance. A brainstem-type LB has a strongly labeled outer ring and a weakly reactive central zone (A). A pale body appears as a moderately or strongly immunolabeled, unevenly stained globule with irregular outlines (B). Large (C) and small (D) irregular masses and fine thread-like structures (E) (perikaryal threads, arrow) lying among the lipofuscin granules in the perikarya are faintly stained. A cortical LB appears as a globular inclusion that has occasional indistinct central lacunae (F). From the substantia nigra (A,B), Edinger–Westphal nucleus (C), raphe nucleus (D), dorsal motor nucleus of the vagus nerve (E), and anterior cingulate gyrus (F), paraffin sections with peroxidase–DAB/hematoxylin. Bar=10 μ for all pictures.

Fluorescence photograph of a representative mixed astroglial–neuronal hippocampal culture. Astroglial cells stain positively for GFAP (rhodamine) and appear red, while neurons are MAP 2+ (FITC) and are green/yellow. Approx. 5–10% of the cells is MAP 2+.

Figure 4.12: Some examples of the Elsevier dataset.

and the best five retrieved images with their caption, including a second image where only the most relevant pixels are displayed. Figure 4.13 will show the retrieval results using text-only for the query '`blood vessel`'.

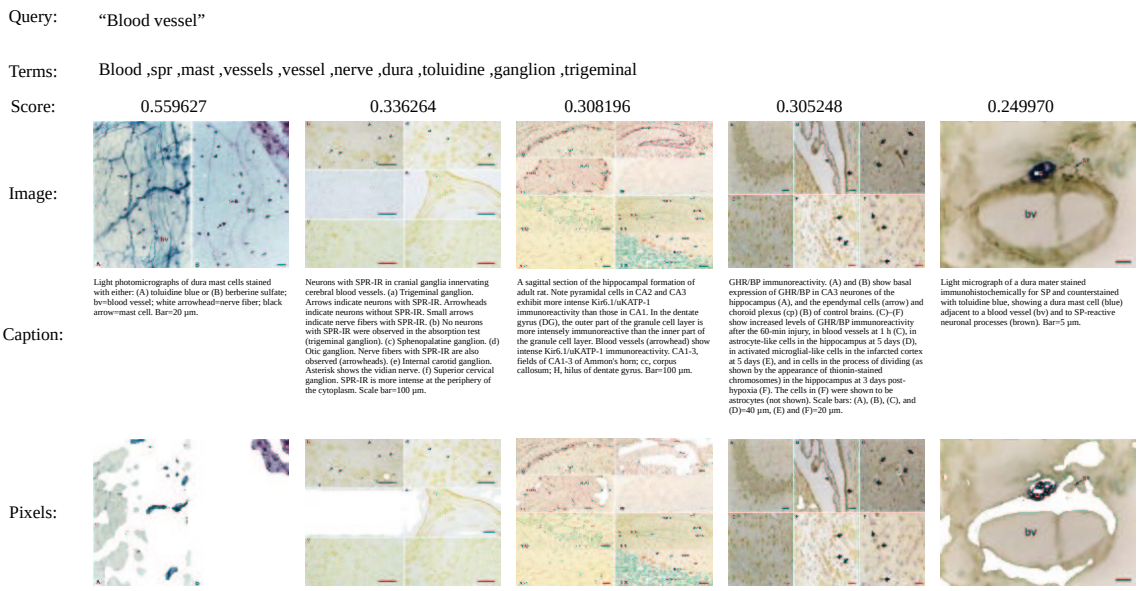


Figure 4.13: Example of text searching for the query '`blood vessel`' in the combined images and text space. The 10 most relevant terms are shown, the score of the retrieved image with caption, the image itself with its caption, and the 50 most relevant image labels are displayed. Non-relevant pixels are left white.

Local image features lend themselves easily for user selection. Letting the user indicate which part of an image is interesting can incorporate relevance feedback. We will now present an example of searching for a part of the first image as found by searching for text only. We used a square block of pixels, however the bag-of-words approach makes any combination of pixels possible. We selected a part of the image where the system found many relevant pixels (upper right corner,

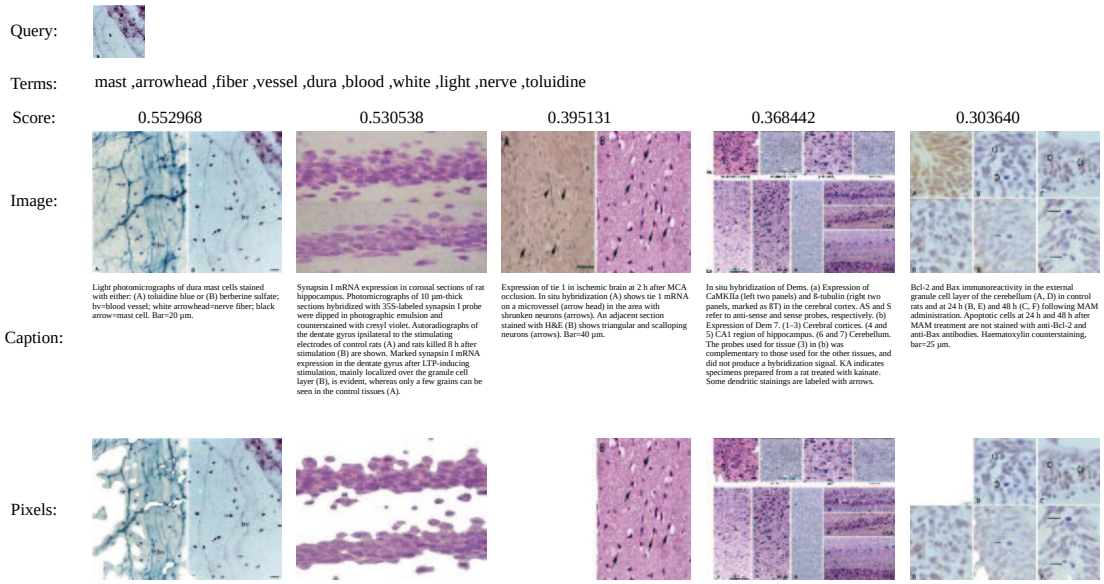


Figure 4.14: Example of image searching in the combined images and text space. The query image is part of the first image found using text only (upper right corner, of the first image in figure 4.13). In the text-only query, this part of the image was indicated as relevant. The 10 most relevant terms are shown, the score of the retrieved image with caption, the image itself with its caption, and the 50 most relevant image labels are displayed. Non-relevant pixels are left white.

of the first image in figure 4.13). Figure 4.14 will show the results when searching with image information only for a selected part of an image.

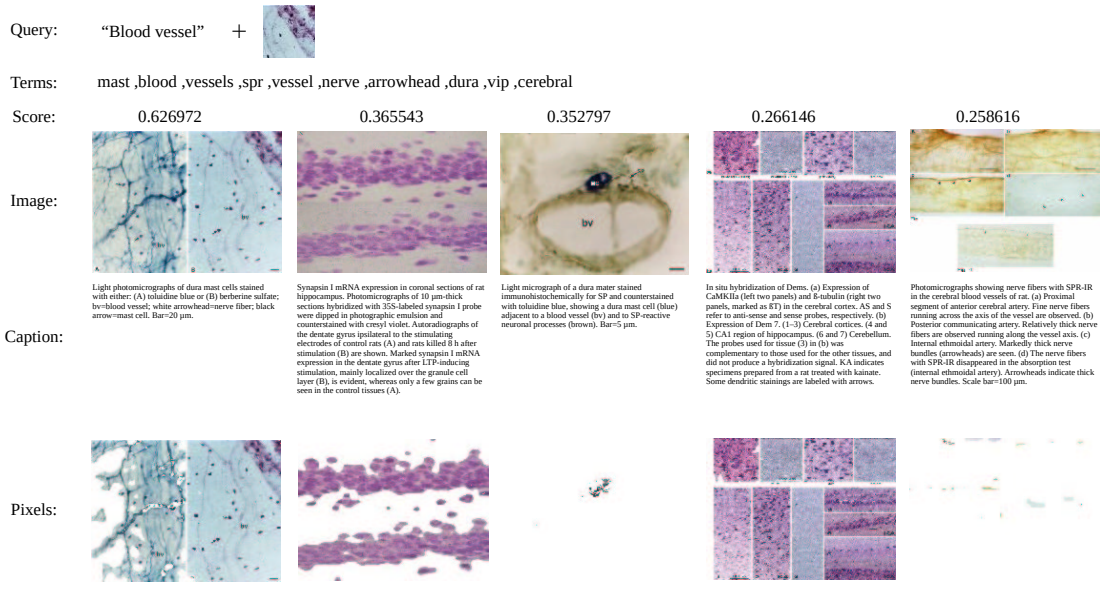


Figure 4.15: Example of image and text searching in the combined images and text space. The 10 most relevant terms are shown, the score of the retrieved image with caption, the image itself with its caption, and the 50 most relevant image labels are displayed. Non-relevant pixels are left white.

Combining both image and text incorporates results relevant to both modalities. Figure 4.15 shows the results when searching with the query '`blood vessel`' and with the selected relevant part of an image found by searching with text only. The results show that images relevant to both the image as the text query are retrieved. Edges are given high weights in the weighing scheme, as they are more informative than uniform image patches. This high weighting is in accordance with work on salient point retrieval [28], where corners and edges are considered highly representative of an image and therefore salient.

Chapter 5

Conclusions and Future Research

This paper presents a new image retrieval scheme and combines text and images for information retrieval. The image retrieval method uses Latent Semantic Indexing on the words in an image. We define the words in an image as a bag of words where each pixel is assigned a label describing the behavior of the image in the pixel. A label stands for a discretized signature of the color N-jet, making up the local spatial and spectral structure of the image in the pixel. Our method solves the problems occurring with blocks of pixels and salient point detection. Blocks of pixels are highly sensitive to specific image conditions. We use rotational invariance and intensity invariants to deal with these problems. Translational invariance is provided by the use of a histogram, and frequently co-occurring structures are mapped to the same concept by the use of LSI. As all pixels are used, this method does not suffer from misdetection of salient points, nor of the sensitivity to a background change. Saliency of pixels is still used, as more informative pixels are given a higher weight than less informative pixels. Furthermore, we use invariant color information, providing powerful clues for image similarity. Local features provide robustness to object occlusion and background changes. As image information is represented in words, the integration of text and image is achieved by adding the words in an image to the words in a text. Using LSI on this combined space learns relationships between image and image, text and text and image and text. Some examples demonstrate the capabilities of the presented approach. When both modalities are presented, one modality tends to describe what the other is missing. Linking images to text can help visualize a concept. Linking text to images adds semantics to images.

More experimental evidence for this method of image retrieval needs to be collected by systematically testing the quality of the proposed method on different datasets. Datasets where images with different and similar shape, texture, size and color are combined provide a richer test environment. Experimental evidence for the link between image and text is needed. Similar to Barnard et al. [1] the words for a given image can be generated and evaluated. Like in Barnard et al. the actual correspondence between which part of an image is linked to which word is much harder to evaluate. For such an evaluation, a specific manual test set needs to be constructed.

Some fine-tuning of the method could be achieved by varying the order of the Taylor series. We use a constant order when using a spatial Taylor approximation. This could change to a variable order approximation depending on the quality of the approximation in a small neighborhood. As we have the original image at hand, we can calculate the precise error of the Taylor series. It

could be useful to approximate an image patch until the approximation error drops below e.g. 10%. This method will use zero or first order approximations for simple uniform patches and use higher order where complex image structures need to be described. Currently the scale σ of the method is fixed, whereas a better performance may be achieved by fine-tuning to a multi-scale approach.

Estimating the LSI dimension remains a problem. Methods as MDL and other measures of model fitting could be used to make an educated guess of a proper dimension. An iterative SVD [6] is a useful extension to our implementation LSI. Our current implementation stores all values in a huge sparse matrix, limiting the vocabulary size to about 70.000 words. Iterative SVD makes much higher vocabulary sizes possible, thus increasing the expressive power of the words in an image. Moreover, evaluating LSI to other learning models will give more insight in the choice of using LSI over other models.

We foresee that the method is easily extendable to include other modalities as audio and video. Any modality can in principle be included, as long as they can be decomposed in 'words' describing the modality locally.

This thesis just scratches the surface of the combination of text and images. It makes some practical relation between words and images. The true relation is much deeper. Humans integrate language and visual information effortless. Small children can already draw a prototypical picture of a concept. Annotating a picture with words is achieved at a very early age. In the very near future, artificial intelligence systems will beat the human chess champion. However, the real challenges of artificial intelligence are the ones we perform so easily.

Acknowledgements

This thesis itself already combines lots of images and lots of words. Finishing the thesis is a combined effort I would like to share with several people. First I would like to thank the unborn child of my supervisor, Jan-Mark Geusebroek. With his wife in a highly expecting state, Jan-Mark superbly passed the pressure on to me. Without Jan-Mark's help, I would probably today still have been working on the thesis, implementing several small improvements to the system and doing very interesting little experiments. Second, I would like to thank Marieke, for believing in me and her never-ending endurance to tolerate me near her, as the theses neared completion. Several other people have helped me in one way or another. An alphabetical listing is found below.

Andy, for some native speaker insights. Arjen, for the reading group et al. Arnold, for his group. Breanndán for fast scheme evalutation. Carla, to help me focus, and not be distracted by small things e.g. Moving. Cees, for the example of your master's thesis. Darek, for finishing his Ph.D. just after my master's. Dennis, for Horus and all his support with it. Didier, for letting me see there is still student mentality in work. Elmer, for still being a 'goede buur'. Elsevier, for letting me use this dataset. Ernst, for his computer, which I didn't use for the thesis, but at least I Could! Family, for making me explain time and time again what I do, therefore increasing my own understanding. Gertjan, for co-evaluating our mutial supervisor. Giang, for our weekends at the office. Isis People, for lunch, coffee, borrels and socials. Jeroen V, for our evenings at the office. Jeroen v. R, for his inspirational sweater. Joanne, for letting me finish my thesis sooner than her's. Joost G, for being the moral and digestive conscience. Joost W, for offering to read the thesis, didn't use that, but at least I Could! Judith, for evenings spent in 'Noord'. Koen, for good times during 'stappen' Leon, for 4 years of nagging about my graduation party. Ma, for inspiration. Maarten, for motivation. Marc G, for chess and beer. Marc W, for setting an example. Marcel, for being in the commission. Marijn, for winning from me with chess. Olaf, for achieving both Quality as Quantity. Pa, for trusting in me. Rein, for letting me learn a lot. Rogier, for Pancakes with Grand Marnier. Rosie, Terrie en de kids, for very 'gezellige' Sunday afternoons. Shirley, for π. Stephan, for our nice chats about (images of) life. Tammo, for 'gezelligeOvenschotelGezelligheid'. Thang, for his suggenstion of the Flag Database I never used, but at least I Could ! Theo, for inspiring talks in the toilet. Thijs, for providing me with the text to Corel I did not yet use, but at least I Could ! To whom I forgot, Sorry.

Jan van Gemert

Bibliography

[1] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research (Special Issue on Machine Learning Methods for Text and Images)*, 3:1107–1135, Februari 2003.

[2] G. Bateson. *Mind and Nature*. Bantam, New York, 1979.

[3] A. B. Benitez and S.-F. Chang. Perceptual knowledge construction from annotated image collections. In *Proceedings of the 2002 International Conference On Multimedia and Expo (ICME-2002) also Columbia University ADVENT Technical Report 001, 2002*, Lausanne, Switzerland, Augustus 2002.

[4] M. W. Berry, T. Do, G. W. O'Brien, V. Krishna, and S. Varadhan. Svdpackc (version 1.0) user's guide. University of Tennessee, April 1993.

[5] D. M. Blei and M. I. Jordan. Modeling annotated data. Technical Report CSD-02-1202, Division of Computer Science , University of California, Berkeley, CA, 2002.

[6] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Seventh Europian Conference on Computer Vision (ECCV)*, volume 1, pages 707–720. Springer Verlag (LNCS 2350), 2002.

[7] M. La Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-based image retrieval on the world wide web. Technical Report 1998-004, Image and Video Computing Group, Boston University, 9, 1998.

[8] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[9] S.T. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, 23(2):229–236, 1991.

[10] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Seventh Europian Conference on Computer Vision (ECCV)*, volume 4, pages 97–112. Springer Verlag (LNCS 2353), 2002.

[11] N. Fuhr. Probabilistic models in information retrieval. *The Computer Journal*, 35(3):243–255, 1992.

[12] L.T.F. Gamut. *Language, Logic and Meaning, Part 2*. Chicago University Press, 1991.

[13] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and A. Dev. Color and scale: The spatial structure of color images. In D. Vernon, editor, *Sixth Europian Conference on Computer Vision (ECCV)*, volume 1, pages 331–341. Springer Verlag (LNCS 1842), 2000.

[14] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. Color invariance. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(12):1338–1350, 2001.

[15] Th. Gevers and A. W. M. Smeulders. Pictoseek: Combining color and shape invariant features for image retrieval. *IEEE transactions on Image Processing*, 9(1):102–119, 2000.

[16] W. I. Grosky and R. Zhao. Negotiating the semantic gap: From feature maps to semantic landscapes. *Pattern Recognition*, 3:593–600, 2002.

[17] P. Grünwald. *The Minimum Description Length Principle and Reasoning under Uncertainty*. PhD thesis, CWI, the Netherlands, march 1998.

[18] D. Hiemstra and A. de Vries. Relating the new language models of information retrieval to the traditional retrieval models. Technical Report TR–CTIT–00–09, Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands, 2000.

[19] T. Hofmann and J. Puzicha. Statistical models for co-occurrence data. Technical Report AIM-1625, International Computer Science Insitute, 1998.

[20] J. J. Koenderink. The structure of images. *Biol. Cybern.*, 50:363–370, 1984.

[21] J.H. Lim. Building visual vocabulary for image indexation and query formulation. *Pattern Analysis and Applications (Special Issue on Image Indexation)*, 4(2/3):125–139, 2001.

[22] C.D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, USA, 1999.

[23] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *Proc. 15th International Conf. on Machine Learning*, pages 341–349. Morgan Kaufmann, San Francisco, CA, 1998.

[24] C. N. Mooers. Application of random codes to the gathering of statistical information, 1948.

[25] Y. Mori, H. Takahashi, and R. Oka. Automatic word assignment to images based on image division and vector quantization. In *RIAO'2000 - Conference Proceedings of the 6th RIAO Conference. Content-Based Multimedia Information Access*, Paris, France, April 2000.

[26] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[27] B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. In *Fourth European Conference on Computer Vision (ECCV)*, volume 1, pages 610–619, 1996.

[28] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, May 1997.

[29] S. Sclaroff, L. Taycher, and M. LaCascia. Imagerover: A content-based image browser for the world wide web. Technical Report 1997-005, Image and Video Computing Group, Boston University, 24, 1997.

[30] C. E. Shannon. A mathematical theory of communication. *belltechj*, 27(3):379–423, July 1948. Continued 27(4):623-656, October 1948.

[31] S. Siggelkow and H. Burkhardt. Image retrieval based on local invariant features. In *IASTED International Conference on Signal and Image Processing (SIP) 1998*, pages 369–373, Las Vegas, USA, October 1998.

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

[33] D. M. Squire, W. Müller, H. Müller, and J. Raki. Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback. In *The 11th Scandinavian Conference on Image Analysis*, pages 143–149, Kangerlussuaq, Greenland, jun 7–11 1999.

[34] R. K. Srihari. Automatic indexing and content based retrieval of captioned images. *IEEE Computer*, 28(9), September 1995.

[35] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[36] T. Westerveld. Image retrieval: Content versus context. In *RIAO'2000 - Conference Proceedings of the 6th RIAO Conference. Content-Based Multimedia Information Access*, Paris, France, April 2000.

[37] X. S. Zhou and T. S. Huang. Unifying keywords and visual contents in image retrieval. *IEEE MultiMedia*, 9(2):23–33, 2002.

[38] L. Zhu, A. Rao, and A. Zhang. Theory of keyblock-based image retrieval. *ACM Transactions on Information Systems (TOIS)*, 20(2):224–257, 2002.