J Intell Inf Syst (2012) 39:651–685
DOI 10.1007/s10844-012-0207-6

Image auto-annotation with automatic selection of the annotation length

Oskar Maier · Halina Kwasnicka · Michal Stanek

Received: 16 November 2011 / Revised: 28 February 2012 / Accepted: 9 May 2012 /
Published online: 26 May 2012
© The Author(s) 2012. This article is published with open access at Springerlink.com

Abstract Developing a satisfactory and effective method for auto-annotating images that works under general conditions is a challenging task. The advantages of such a system would be manifold: it can be used to annotate existing, large databases of images, rendering them accessible to text search engines; or it can be used as core for image retrieval based on a query image's visual content. Manual annotation of images is a difficult, tedious and time consuming task. Furthermore, manual annotations tend to show great inter-person variance: considering an image, the opinions about what elements are significant and deserve an annotation vary strongly. The latter poses a problem for the evaluation of an automatic method, as an annotation's correctness is greatly subjective. In this paper we present an automatic method for annotating images, which addresses one of the existing methods' major limitation, namely a fixed annotation length. The proposed method, PATSI, automatically chooses the resulting annotation's length for each query image. It is held as simple as possible and a build-in parameter optimization procedure renders PATSI de-facto parameter free. Finally, PATSI is evaluated on standard datasets, outperforming various state-of-the-art methods.

Keywords Image auto-annotation · Variable annotation length · Parameter optimization · Image similarity

This research has been partially supported by the Polish National Centre for Research and Development under grant SyNaT.

O. Maier (✉) · H. Kwasnicka · M. Stanek
Institute of Informatics, Wrocław University of Technology, Wrocław, Poland
e-mail: oskar.maier@googlemail.com

H. Kwasnicka
e-mail: halina.kwasnicka@pwr.wroc.pl

M. Stanek
e-mail: michal.stanek@pwr.wroc.pl

1 Introduction

Images are a common element of everyday life in our visually oriented world. They accompany other information bearers or speak for themselves. Appreciated for their expressiveness and aesthetic values, they are one of the most powerful means of transporting information—"A picture is worth a thousand words", to quote a popular proverb. No technique could take their role in inter-human communication, no computer grasp the dense, highly subjective information encoded in a picture. As a complex idea can be conveyed with just a single still image, they're made use of and sought for in all areas where communication is essential.

To utilize their power, suitable images must be found that transport the intended meaning. This often requires tedious and time-consuming browsing of large image libraries. In the more narrow domain of a highly specialized archive the required effort might be acceptable, but normally it is not. The widest domain generally accessible and at the same time the most unstructured is the Internet. Manual browsing is here practically impossible. As the size of digitally available image stocks continuous to grow and their usage becomes more and more frequent and wide-spread, the need of efficient, reliable and robust image retrieval solutions becomes more pressing.

Traditional Internet image search engines deploy Text-Based Image Retrieval (TBIR), an approach relying on meta-data. Meta-data, gained for example by examining the text surrounding an image embedded in a document, is used to establish a relevance measure for a textual search query. Well explored, with fast indexing and retrieval architectures speaking for it, this method suffers from the often low semantic coherence between the image and the surrounding text. Furthermore TBIR cannot work with images for which no or only weak meta-data is available.

Content-Based Image Retrieval (CBIR) aims towards more relevant search results by analysing the actual image content and lifting the exclusive dependence on meta-data. One approach is query-by-example, where a query image is presented to the search engine which then produces a number of similar images. Another proposition is semantic retrieval which aims to name semantic concepts appearing in an image. This can be either achieved by directly associating each semantic term of the query with an explicit object or class in the images or to annotate the images with a number of relevant keywords. The first, straightforward approach would require to design and implement an algorithm for each distinguishable semantic term. Beside the method's tremendous complexity, a semantic term is an abstract class and the concrete objects associated with it differ not only between cultures but also individuals (see Eakins et al. (2004) for an interesting discussion on subjective perception and user needs in image retrieval). This proves to be an insurmountable problem.

The second approach utilizing image annotation is more practicable. Therefore our study concerns CBIR on the base of annotations. The main contributions of the automatic image annotation method proposed in this paper (called PATSI—**P**hoto **A**nnotation **T**hrough **S**imilar **I**mages) are (i) it automatically chooses the length of an annotation assigned to a query image and (ii) its parameters are automatically optimized. The PATSI method's main advantages in comparison to other known approaches are specified in the next section.

This paper is organized as follows. The next section presents related approaches and underlines the PATSI method's important novelty. Section 3 describes our annotation method with detailed explanations of its phases, the proposed transfer

functions and distance measures plus how it handles the automatic image annotation challenges. A short discussion on the evaluation measures is provided in Section 4. Next, in Section 5, the parameter optimization method is described in detail. Section 6 presents experimental results where we compare results obtained from our method to other existing state-of-the-art annotation methods. Finally we discuss the results in Section 7, summarize PATSI's advantages and weaknesses and give an outlook to the future.

2 Related approaches

CBIR can be approached by annotating images, in which case a few relevant keywords are assigned to each image to constitute its meta-data. Such the image content is taken into account, while the highly developed TBIR search engines can still be employed. Drawbacks are firstly that only a few semantic terms are used to describe the whole image, which does not begin to do the rich expressiveness justice. Secondly a term often refers only to a limited area, while being assigned globally to the picture. This leads to an often weak correspondence between keywords and image content. Nevertheless, the results's relevance can be expected to increase significantly compared to traditional TBIR and the challenges connected with this approach are conquerable, rendering it the most chosen path in the recent years.

Labels, i.e. collections of semantic terms attached to an image, can be created manually. This way human perception of the images characteristics is well represented and explicit, speaking labels are created. But the tedious, costly and often error-prone process of manual labelling renders it unsuitable for all but the smallest, seldom extended collections. A method to automatically assign keywords to images is required. Achieving this through direct, explicit object recognition is mostly inapplicable for the reasons mentioned above.

Treating automatic image annotation as an classification task for every semantic term from a closed set introduces machine learning to the problem. Large image collections, often weakly annotated training data (Carneiro et al. 2007) and the always present semantic gap, characterizing the difference between the visual and textual object representations, present themselves as major challenges.

The majority of studies in the automatic image annotation field use machine learning techniques to learn statistical models from annotated images and apply them to generate annotations for unseen images. These methods can be divided into two main categories: probabilistic modelling methods and classification methods. Some interesting methods belonging to the first category are: Hierarchical Probabilistic Mixture Model (HPMM) by Hironobu et al. (1999), Translation Model (TM) by Duygulu et al. (2002), Supervised Multi-class Labelling (SML) by Carneiro et al. (2007), Continuous Relevance Model (CRM) by Lavrenko et al. (2003), and Multiple Bernoulli Relevance Models (MBRM) by Feng et al. (2004). The last is an extension of CRM. Based on the Bernoulli Relevance Models, it outperforms the other methods as reported in Feng et al. (2004). CRM by Lavrenko et al. (2003) and MBRM by Feng et al. (2004) are used as a reference baseline by many researchers working in the image annotation area. Methods of the second category employ trained classifiers to find correlations between the words and the annotated image's visual features. We can mention here Bayes Point Machine (Chang et al. 2003), Support Vector

Machine (Cusano et al. 2004) and Decision Trees (Kwasnicka and Paradowski 2008) which all estimate the visual features distributions associated with each word. Some authors try to refine the annotation results by reducing the difference between the expected and resulting word count vectors (Kwasnicka and Paradowski 2006), by using Word-Net which contains semantic relations between words (Jin et al. 2005) or by word co-occurrence models coupled with fast random walks (Llorente et al. 2009), an interesting approach exploiting the recent advances in graph processing.

A family of baseline methods proposed by Makadia et al. (2008) share the assumption that visually similar images are likely to share the same annotations. The annotation process then relies on transferring labels from a number of the nearest neighbours.

The weighted nearest neighbour model was adopted by Verbeek et al. (2010) in the TagProp image annotation system. Keyword relevance in TagProp is predicted by taking a weighted sum of the most similar images' annotations in an annotated training set. Verbeek et al. (2010) propose an optimization process which allows to obtain a distance between visual features that corresponds to the textual distance between annotations.

The simple method presented in Makadia et al. (2008) outperforms most of the more complex approaches and questions the need for sophisticated algorithms. Makadia method's drawback is on the one hand the dependency on manually set parameters, such as the neighbourhood size, and on the other hand the annotation length's restriction to a fixed, predetermined size. Both influence the resulting annotations quality. Particularly the latter directly effects the quality measures: in general shorter annotations lead to higher precision and lower recall and vice versa (see Section 4 for a short discussion on the evaluation measures).

An annotation transfer step very similar to Makadia et al. (2008) has been recently proposed by Medvet et al. (2011) to assign text-mined names to faces. They extend the original approach by weighting the considered name based on the nearest images distance to the query image and then transfer the name only if the weight surpasses a threshold.

We propose a simple method for **P**hoto **A**nnotation **T**hrough **S**imilar **I**mages (PATSI) based on the hypothesis that similar images should share a large part of the annotations inspired by Makadia et al. (2008). We incorporate the nearest neighbour approach and keep our method as simple as possible. Contrary to them we also address the problems occurring from a fixed annotation length by using weighted annotation and selecting terms with weights values greater than a threshold for transfer. We select a more general and simple transfer function than Makadia et al. (2008)'s transfer schema and such following Medvet et al. (2011)'s proposition. For further simplification we use only a single distance measure and a single feature set. Our method is designed towards solving the problem of choosing the assigned annotation's appropriate length—it is our methods's main advantage and novelty, neither addressed in Makadia et al. (2008) (who use fixed annotation length) nor in Medvet et al. (2011) (who face only a binary decision problem). Additionally we propose a transfer parameter optimization method which automatically tunes the resulting word count associated with the image, eliminating the need of manually setting the parameters, and such rendering our method de-facto parameter free. On top of this we investigate a number of interesting and representative transfer functions and PATSIs sensitivity to a variety of distance measure and feature sets.

High accuracy obtained by the proposed approach on the standard benchmark image datasets in conjunction with the method's simplicity and generality and its computational efficiency make it a perfect candidate for a baseline in the field of automatic image annotation.

3 Automatic PATSI with variable annotation length

Automatic image annotation is the task of assigning a number of words w from a closed *vocabulary* $W = w_0, ..., w_k$ (sometimes also referred to as *dictionary*) to a formerly unseen query image Q which meaningfully describe the semantic concepts in Q. All assigned words together form the new annotation a_Q of Q, where each word $w_j \in W$ is a natural language term naming a semantic concept. The goal is to obtain an informative description of Q that ideally conveys all (relevant) semantic concepts present in the image. This formulation leads directly to the main challenge that automatic image annotation, image retrieval, possibly the whole field of computer vision (according to Tousch et al. 2012) faces: the semantic gap. Defined by Smeulders et al. (2000) as the 'lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data has for a user in a given situation', the semantic gap names the struggle to model the barely understood human visual perception[1] (Michalak et al. 2011). The task is further complicated by the high subjectivity of an annotations relevance. The importance of a semantic concept present in an image, how to call it and at which abstraction level to name it cannot be solved universally, but depends on the situation and the user as well as his cultural background.

Researchers show a tendency to take a pragmatic approach towards the semantic gap and mainly concentrate on developing some practical albeit domain-limited solutions. Popular approaches are the learning and modelling of word to image (or word to image region) connections and including the user in the retrieval loop (see Smeulders et al. (2000); Datta et al. (2008) for two excellent surveys).

User interaction is impractical to implement into PATSI, as automatic image annotators have to work unsupervised. The proposition to build word-models or train classifiers is popular but faces the problem that which part of an image a keyword describes is generally unknown and hard to discover (Datta et al. 2008) despite advances in automatic image segmentation (e.g. colour segmentation used in Aoun et al. (2011), EM-algorithm-based in Carson et al. (2002), normalized cut criteria in Shi and Malik (2000)) or implicit relation modelling (e.g. Carneiro et al. 2007). Recently another approach has been suggested by Makadia et al. (2008) and picked up by Medvet et al. (2011), that we adopted and extended in this paper.

PATSI is build on the assumption that similar images share large parts of their annotations (Stanek et al. 2010a, b). Such we essentially perform image retrieval by example with a subsequent annotation transfer step. We rely on the transfer method to punish outliers and promote often appearing keywords (see Section 3.3 for an example).

[1] Not just the mechanical part, but with all its implications and subjectivity.

It remains the problem of defining image similarity for the retrieval task. We decided to employ only simple features and distance measures to emphasize the performance of our annotation transfer. We performed a number of experiments comparing different features and distance measures, whose descriptions and results can be found in Section 6. The good results obtained support our decision.

According to Datta et al. (2008) most CBIR systems perform feature extraction in a preprocessing step. The same holds for PATSI, which can be divided into a learning phase, in which the annotators knowledge is assembled, and a processing phase of image retrieval and annotation transfer. Both are laid down in detail in the remainder of this section. In Section 3.3 we describe how PATSI handles the problem of weakly annotated databases and in the last Section 3.4 we summarize our contributions.

3.1 Learning phase

In the learning phase, knowledge is extracted from the collection of manually annotated images in the training set D. A number of features are extracted from each image $I_x \in D$ and combined into a feature vector F_x. The set K of tuples (F_x, a_x) containing the feature vectors and the associated manual annotation constitutes the annotator knowledge.

The learning phase is only executed once, but can be repeated if the annotator should be adapted to a new or significantly altered training set. As a novelty in PATSI, the learning phase in succeeded by an additional parameter optimization step presented in Section 5 and rendering our method de-facto parameter-free.

3.1.1 Image features

Features are image descriptors that convey relevant information. The ideal feature:

1. detects and highlights all landmarks that serve the detection of and differentiation between semantic concepts,
2. compresses, i.e., is by magnitudes smaller than the images pixel representation and
3. does not loose any relevant information in the process.

These three requirements often contradict each other and are very hard to satisfy. No information about the task at hand (What is sought?) and/or the user (Who seeks?) is available at feature extraction time to determine what is relevant and what not. This arises from the semantic gap discussed above. Furthermore different keywords (e.g. *mountain*; *merry*) describe different concepts (e.g. *landmark*; *mood*), each of which is composed of a number of criteria (e.g. *colour*, *shape* and *position*; *facial expression* and *situation*). It is highly unlikely for a single feature to capture the numerous semantic concepts, except for the most restricted vocabularies.

Popular features in image retrieval include *colour* (Chatzichristofis and Boutalis 2008a; Goodrum 2000; Huang et al. 1997), *texture* (Chatzichristofis and Boutalis 2008b; Zhang et al. 2000; Haralick et al. 1973), *interesting points* (e.g. Lowe 2004) and *shape*. They can be categorized by which image part they are extracted from. Usual approaches are, *global*, *local* from a regular grid and *local* from presegmented image's regions (Mikolajczyk et al. 2005).

Our focus lies on simplicity, so we refrain from complex pre-segmentation. Instead we concentrate on simple and often employed features extracted either globally or on a local grid.

We perform an investigation of PATSI's performance in conjunction with various features and feature-combinations. Their description including relevant references can be found in Section 6.3. A detailed discussion of their characteristics is beyond the scope of this work. The interested reader is referred to the comparison papers Deselaers et al. (2008a); Grigorescu et al. (2002).

3.2 Processing phase

While PATSI's learning phase is similar to many methods in literature, the processing phase contains some novelties. It is triggered by presenting a query image to the annotator. Figure 1 shows this phase's graphical representation.

In the first step the query image Q (Fig. 1 step ①) is processed to extract the feature vector from this image. Then a *distance measure* is applied to compute the distances between a query image and all images from the data set. The *k nearest neighbours* (i.e., the k images nearest to the query image) are kept, the others are dismissed. This set of size k is sorted in ascending order by distance i.e., the rank 1 is assigned to the nearest image, rank k to the furthest one.

Now a *transfer function* is applied to the keywords associated with each of these selected k images. Taking into account rank and/or distance to the query image, the transfer function assigns to every keyword a numeric value called transfer value (*t-value*). The same keywords' transfer values are summed up, thereby the keywords appearance's frequency is taken into account. Finally, every keyword whose transfer value passes a *threshold t* is transferred to the query image. The resulting annotations length is not fixed, but depends on the selected threshold value t and varies for each query image. In this our approach differs from known methods such as Makadia et al. (2008); Carneiro et al. (2007) or even Medvet et al. (2011) who also employ transfer functions.

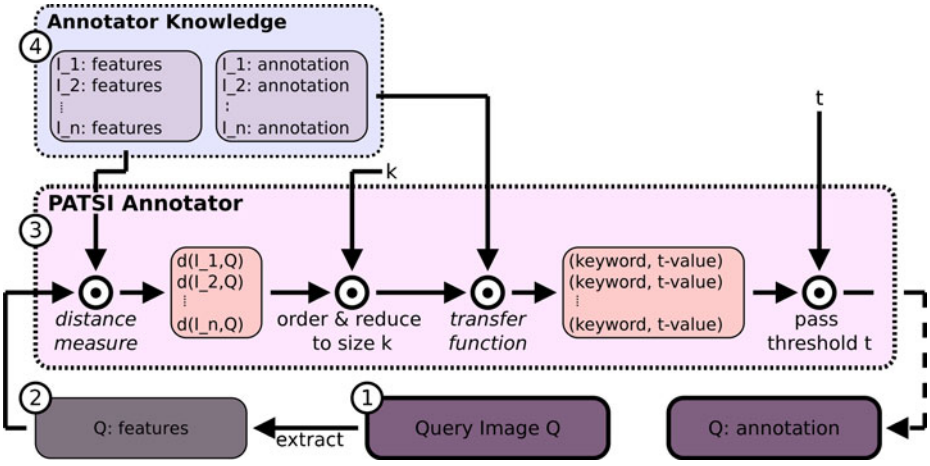


Fig. 1 Graphical representation of PATSIs processing phase

In a post-processing step the query image features and annotation can be added to the annotator knowledge. This increase of the annotator knowledge does not lead to any dictionary change and therefore the learning phases does not need to be repeated.

PATSI's performance depends on the two parameters k and t, which values have to be carefully tuned. Their optimal values not only depends on the features/distance measure combination employed, but also on the training dataset. The majority of methods from literature (e.g. Makadia et al. 2008; Carneiro et al. 2007; Medvet et al. 2011; Wichert 2008; Boccignone et al. 2008) depend on one or more parameters which are manually tuned to the task at hand. As a step towards overcoming this limitation and towards a working system, we present an optimization method for the PATSI parameters in Section 5.

3.2.1 Distance measure

Similarity between images is an abstract concept. The human perception of similarity involves subjectivity and its function is largely unknown. Mathematical similarity, on the other hand, is clearly defined and restricted. A distance measure $d(\cdot)$, the inverse of a similarity measure, is a positive function such that its value is greater when two images are farther apart. $d(I_x, I_y)$ between any two images I_x, I_y has to meet the following conditions that define a metric:

$$d(I_x, I_y) = 0 \Longleftrightarrow I_x = I_y, \tag{1}$$

$$d(I_x, I_y) = d(I_y, I_x) \tag{2}$$

$$d(I_x, I_z) \leq d(I_x, I_y) + d(I_y, I_z) \tag{3}$$

That is, it yields the value 0 only for identical images, is symmetrical and satisfies the triangle inequality. These conditions imply that for all pairs of different images $d(I_x, I_y)$ returns a positive value:

$$d(I_x, I_y) > 0 \Longleftrightarrow I_x \neq I_y \tag{4}$$

A similarity function essentially maps a pair of images to a real value.

PATSI does not compute the distances between the images directly, but rather over their feature vectors F_x and F_y.

Mathematical models strive to capture the human subjective similarity perception, but the semantic gap forces a discrepancy between the computed and personally perceived similarity. In our experiments we compare a number of popular distance measures which are described in Section 6.3, to determine which serve best to narrow the gap.

3.2.2 Similarity space

Choosing the feature set and the distance measure is not independent and some combinations are not feasible. For a more compact notation and to emphasize the their strong connection, we refer to the conjunction of a feature set and a distance measure as *similarity space* (SS). The SS used in an annotator has to be chosen carefully, as this decision greatly effects the resulting annotations' quality. In Section 6.3

we therefore provide an exhaustive comparison of different feature set/distance measure combinations.

The current version of PATSI does not allow the combination of different distance measures. It would be interesting to further investigate in this direction, especially as Makadia et al. (2008) report a significant gain in F-score with their *Joint Equal Contribution* method. Some ideas on how to combine different SSs into one transfer process are presented in the conclusion Section 7.

3.2.3 Transfer function

Our goal is to predict an unseen query image's keywords, on the basis of annotations found in the image neighbourhood. The classical k-nearest neighbour classification is frequently used (Makadia et al. 2008; Medvet et al. 2011): the frequency of a keywords appearance in the query image neighbourhood decides about whether or not it is transferred. In PATSI we incorporate information about the relative distances between the query image and neighbours by assigning weights. This way keywords describing the near images, contribute more than the ones of remote images. To calculate the weight we use a defined *transfer function* (TF). In this paper we concentrate on two classes of possible TFs that we believe to be the most promising: *distance based* and *ranked based*. The first utilizes directly the distance between images, while the second makes use of the rank ranging from 1 to the neighbourhood parameter k.

From the class of the distance based TFs we select the following four for further investigation:

$$t^1(Q, I) = \frac{1}{d(Q, I)}, \tag{5}$$

$$t^2(Q, I) = \frac{1}{d(Q, I)^2}, \tag{6}$$

$$t^3(Q, I) = e^{-d(Q,I)}, \tag{7}$$

$$t^4(Q, I) = e^{-d(Q,I)^2}, \tag{8}$$

where $d(Q, I)$ is a distance value measured between the query image Q and one of its neighbours I. The TFs plots are displayed in Fig. 2.

In function t^1 (Fig. 2a) the weights assigned to the keywords are inversely proportional to the distance between the query and the neighbouring images. Both t^1 and t^2 put the highest emphasis on the nearest neighbours, while minimizing the impact of further ones. The smoothed descend of t^3 leads to more weight for distant images' annotations. The same is true for the TF t^4, which additionally treats the nearest neighbours almost equally. This can balance out small errors in the distance measure. The range of values returned by this class of TFs is highly dependent on the applied distance measure and therefore sensitive to parameter tuning.

With ranked based TFs we propose a second family of TFs which give more deterministic weights, as their range depends only on the value of k. Such they are more robust against changes of t and k. The rank based TFs are inspired by Makadia

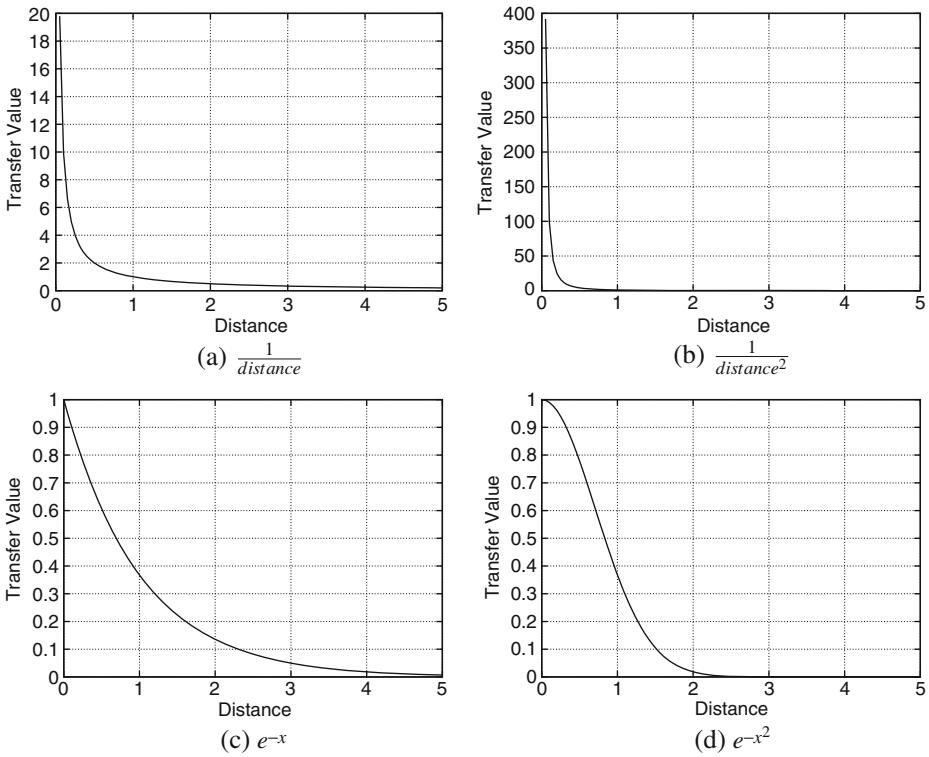


Fig. 2 Distance based transfer functions

et al. (2008)'s findings (presented in their work in Table 3b), who evaluated the individual neighbour images in isolation resulting in a nearly linear drop in F-score with increasing image rank. Within this class we propose four transfer functions:

$$r^1(Q, I) = \frac{1}{rank(Q, I)}, \tag{9}$$

$$r^2(Q, I) = \frac{1}{rank(Q, I)^2}, \tag{10}$$

$$r^3(Q, I) = \frac{(k+1) - rank(Q, I)}{k}, \tag{11}$$

$$r^4(Q, I) = \frac{(k+1)^2 - rank(Q, I)}{k^2}, \tag{12}$$

where $rank(Q, I)$ describes the rank of image I in the query image Q's neighbourhood, ordered increasingly by the distance values $d(Q, I)$; k is the neighbourhood's size (a method parameter). The discrete function graphs are displayed in Fig. 3.

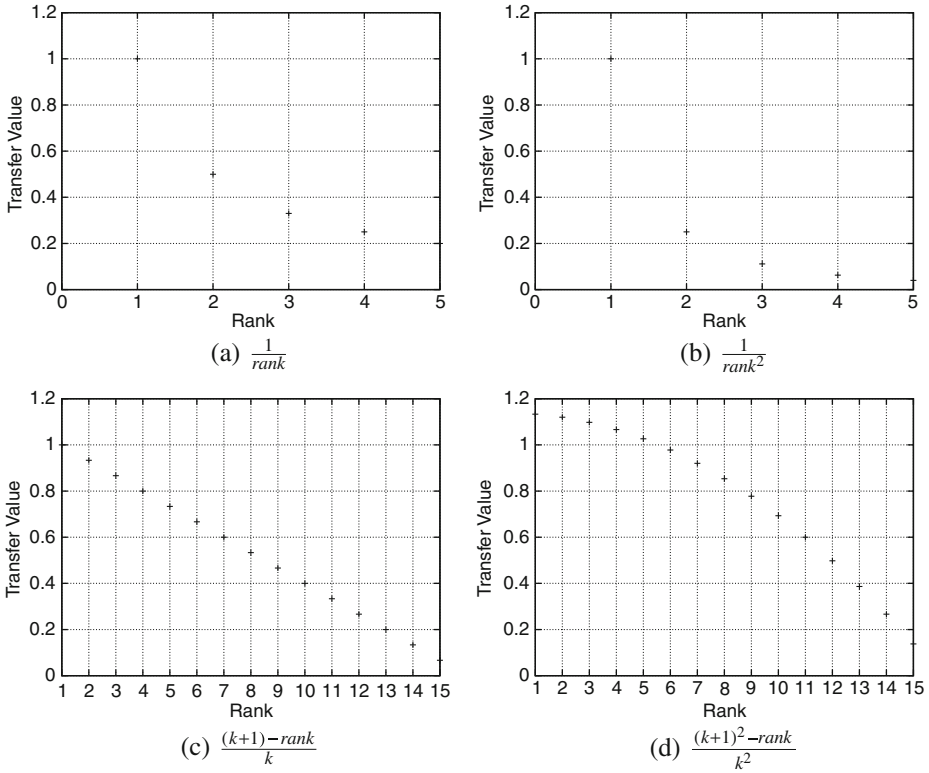


Fig. 3 Ranked based transfer functions

TFs r^1 and r^2 do not take the neighbourhood size k into account and assign fast decreasing weights with higher rank. r^3 is linearly decreasing, while r^4 assigns sufficiently high weights even to high ranked images for them to contribute to the query image Q's resulting annotation. r^1 is the TF most similar to the nearest images' individual scores observed by Makadia et al. (2008) and indeed performed best in the experiments.

Our distance based TF t^1 is equal to the one employed by Medvet et al. (2011). None of our TFs is directly comparable with Makadia et al. (2008)'s approach, as their TF does not assign weights to the annotation word, but rather establishes a ranking from which the first n keywords are selected. Two factors determine their ranking: (i) the local frequency (which in our notation can be represented as a TF with a fixed return value of 1 independent of ranking and distance) and (ii) the keyword co-occurrence in the training dataset. The TFs proposed by us are more flexible and simpler, as they do not access knowledge form the training database. Interestingly Makadia et al. (2008) also report having evaluated t^1 as a TF, but dismissed it due to inferior results.

3.3 Weakly annotated databases

Automatic image annotation faces the problem of weakly labeled databases. Following Carneiro et al. (2007) a training set is weakly labeled if (i) the absence of a word in an image's annotation does not necessarily mean that the connected semantic concept is not present in the image, and (ii) the image's region which corresponds to a word in the image's annotation is not known. PATSI's proposed transfer step is implicitly addressing both of these problems. This is best explained with an example. Figure 4 shows images of an annotation transfer. The query image (Fig. 4a) and three of its nearest neighbours (Fig. 4b–d) are displayed. The first problem can be observed in the third neighbours annotation, from which the word *sky* is missing while the semantic concept is clearly present. But since PATSI takes the whole neighbourhood of k images into account, it is very likely that the annotation word appears in the remaining neighbours' annotations as indeed in this example. How PATSI treats the second problem is also visible in this example. The semantic concept *sky* is normally associated with the pixels in the images's upper part. Our method has no explicit knowledge of this relation, but in the k nearest neighbours the associated annotation word *sky* is likely to appear in a number of images' annotations. Since each appearance of *sky* contributes to its weight, it will pass the threshold t and be transfered to the query image. The other annotation word present in the neighbour images like *grass*, *tree*, etc. will appear more rarely and subsequently will not pass the threshold t.

3.4 Summary

As in Carneiro et al. (2007)'s *Supervised Class Labeling* we represent images as collections of independent feature vectors. But contrary to them we do not use these to build a probability model in the learning phase. Instead we follow the simpler method of Makadia et al. (2008) in treating automatic image annotation as an image retrieval problem with an additional final step of annotation transferring. We profit from their approach's algorithmic and computational simplicity while addressing their most pressing shortcomings, namely (i) the fixed parameter length and (ii) their method's sensitivity to the parameters k and n, by introducing a threshold based annotation transfer system.

(a) Query image: *sky* (b) Neighbour 1: *sky, grass,tree* (c) Neighbour 2: *sky, ocean* (d) Neighbour 3: *clouds, snow, person*

Fig. 4 Example of annotation transfer through shared similarity. The word *sky* appears in two of the neighbour images and is therefore likely to pass the threshold t and to be transfered to the query image. The other annotations receive only marginal weights and are subsequently not transferred

While our proposition such solves the problem of the fixed annotation length, we still depend on the neighbourhood size k and the threshold t. Hence, as second novelty, we propose parameter optimization algorithm to determine near-optimal values for both k and t. This approach is described in detail in Section 5.

To investigate PATSI's modularity and quality, we conducted an exhaustive number of experiment. In Section 6.1 various TFs are tested, Section 6.3 compares a large number of SSs and in Section 6.4 we evaluate PATSI against a number of state-of-the-art algorithms from literature on freely available benchmark datasets where it lies equal with the best methods.

4 Evaluation measures

To rate the quality of our method and to compare it to others we require an evaluation measure. With precision and recall we use two well-known and often employed (Makadia et al. 2008; Carneiro et al. 2007; Nowak et al. 2011; Nowak and Huiskes 2010; Stanek et al. 2010a) measures of annotation relevance. Precision is the fraction of all retrieved words that are relevant, while recall is the fraction of relevant words that are retrieved. They are defined as

$$\text{precision} = \frac{|\text{relevant} - \text{words}| \cap |\text{retrieved} - \text{words}|}{|\text{retrieved} - \text{words}|}$$

and

$$\text{recall} = \frac{|\text{relevant} - \text{words}| \cap |\text{retrieved} - \text{words}|}{|\text{relevant} - \text{words}|}$$

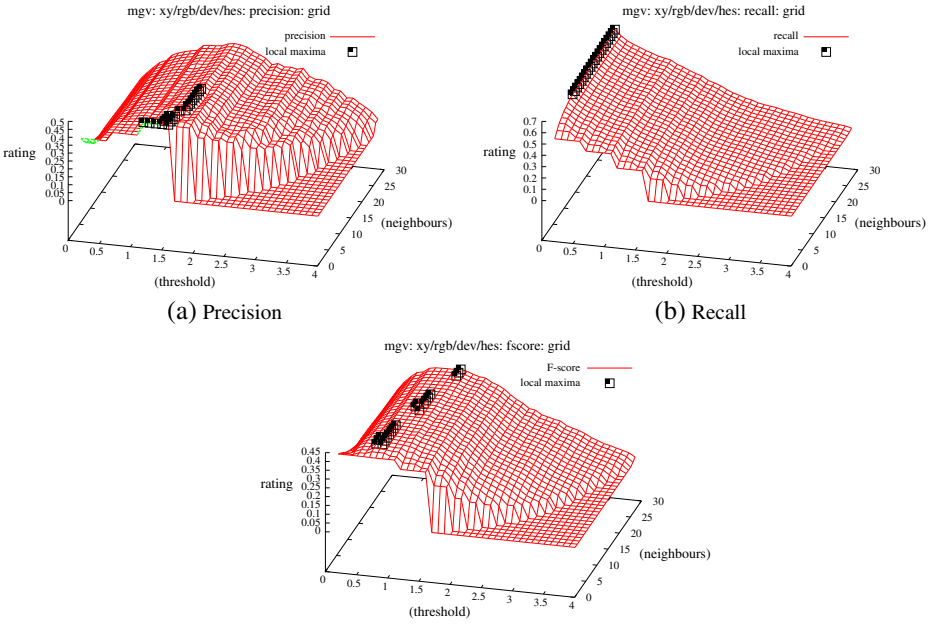
A maximum precision value means no false positives, while a maximum precision value means no false negatives. Combined into the F-score, their harmonic mean, they constitute a meaningful measure of an annotations quality against a ground truth

$$\text{F} - \text{score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

It should be pointed out, that these metrics are not ideal. As described in more detail by Tousch et al. (2012) they measure the gap between an ideal ground-truth and the actual annotation, counting errors between categories without taking their semantic relationship into account. A confusion between *man* and *woman* weights the same as between *man* and *aeroplane*. Furthermore synonyms such as *cop* and *policeman* and categories such as *man*$\subset$*human* are not taken into account. Last years ImageClef challenge (Nowak et al. 2011) is among the first regarding the vocabularies taxonomy in their evaluation, in their case the Flicker Tag Similarity was used to denote similarity between words.

5 Parameter optimization

The quality of an annotation produced by PATSI for a given transfer function and similarity space depends on the number of neighbours k taken into account as well as on the threshold value t. The graph in Fig. 5 details this dependency by showing



(a) Precision (b) Recall

(c) F-score

Fig. 5 Comparative graphs of precision, recall and F-score quality measures on MGV 2006 database

the annotation quality (measured once by each precision, recall and F-score) as a function ϕ of k and t.

Unfortunately, this function's shape varies strongly for different choices of TF and SS. A general solution for $\phi(t, k)$ that is valid for all combinations of TFs and SSs cannot be found. $\phi_{TF,SS}(t, k)$ has to be optimized for a fixed choice of a TF and a SS. We propose an optimization method that yields not the optimal, but sufficiently good values k^* and t^* for a given choice of TF and SS. It can be executed in the training phase and such is not increasing the processing phase's computational complexity.

Finding k^* and t^* can be treated as an optimization problem over $\phi_{TF,SS}(t, k)$ in a time-critical environment and proves to be a non-trivial task. $\phi_{TF,SS}(t, k)$ is a non-linear dependency with a number of local maxima and a discrete domain of k. Furthermore, the annotation process' evaluation is very costly and should therefore be executed very rarely. Linear solvers are not applicable, non-linear solvers cannot work with the discrete k domain. This could be overcome by a prior approximation of $\phi_{TF,SS}(t, k)$, but tests have shown that it leads to an unacceptable high error. A standard brute-force approach requires too many calls to the transfer annotator. To reduce this number we propose the *iterative refinement* algorithm. Figure 6 presents the method's schema.

The search for the near-optimal t^* and k^* consists of three steps:

1. Initially a broad, regular grid is laid over the search space (the function's surface in Fig. 5c) and $\phi_{TF,SS}(t, k)$ is queried for each point where the grid lines intersect (visualized in Fig. 6 by big crosses).

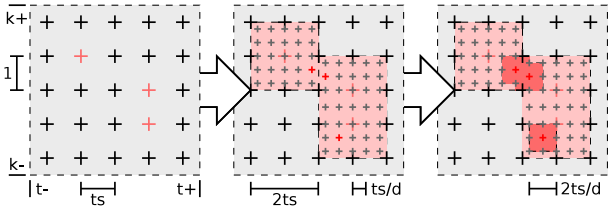


Fig. 6 Iterative refinement schema

2. The points for which $\phi_{TF,SS}(t,k)$ yielded the highest F-scores from the previous step are used as centers for a number of new, smaller regions. A finer grid than in the step before is superimposed on these areas and again $\phi_{TF,SS}(t,k)$ is triggered for the intersection points.
3. If the stop condition is not reached, repeat step 2, otherwise terminate and return the (t,k) pair that produced the highest F-score as the near-optimal solution t^* and k^*.

The notation used in Fig. 6 corresponds to the variables of the formal Algorithm formulation 1. Initially the optimization method requires a number of parameters:

– The initial region on which to search the optimal values, defined by its boundaries k_- to k_+ and t_- to t_+.
– The grid's initial granularity along the threshold axis t_s.
– The *divider* which determines the grid's refinement in each iteration step.

Algorithm 1 PATSI optimization with iterative refinement algorithm

Require: t_s – initial threshold step
k_- – minimal neighbours bound
k_+ – maximal neighbours bound
t_- – minimal threshold bound
t_+ – maximal threshold bound
M – number of interesting areas to further investigate
divider – threshold grid's refinement in each step
stop_condition – stop when minimal improvement is less than this

1: Prepare points on the grid
$P = \{(t,k) | k_- \leq k < k_+ \wedge k - k_i \mod 1 = 0 \wedge t_- \leq t < t_+ \wedge t - t_- \mod t_s = 0\}$
2: **repeat**
3: $S = \{(k,t,\phi(k,t)) | (k,t) \in P\}$
4: obtain S^* where
$S^* \subset S \wedge |S^*| = M \wedge \nexists_{S' \subset S}. \sum_{(a,b,c) \in S^*} c < \sum_{(a',b',c') \in S'} c' \wedge |S'| = |S^*|$
5: $P = \bigcup_{(k',t',c') \in S^*} \{(k,t) | k' - 1 \leq k \leq k' + 1 \wedge k - k' + 1 \mod 1 = 0 \wedge t' - t_s \leq t \leq t' + t_s \wedge t - t' + t_s \mod \frac{t_s}{divider} = 0\}$
6: $t_s = \frac{t_s}{divider}$
7: $\epsilon = |\max_c ((k,t,c) \in S^*) - \min_c ((k,t,c) \in S^*)|$
8: **until** *stop_condition* $> \epsilon$

– The number of each rounds maxima M to consider in the next iteration step.
– A *stop_condition* causing the algorithm to terminate if the last iterations gain was less than this value.

Firstly an initial set of points P is collected over the grid of granularity $t_s \times 1$ with the boundaries k_- to k_+ and t_- to t_+. $\phi_{TF,SS}(t,k)$ is evaluated once for each point $(k,t) \in P$ to form the set S of triples $(k,t,\phi(t,k))$. Next we extract the subset S^* from S, containing only the M triples with the highest values for $\phi(t,k)$. These points constitute the regions to investigate in the next iteration step. A finer grid is superimposed on the M new areas and new sets of P, S and S^* are computed as before. These steps are repeated until the relative improvement ϵ is lower than the *stop_condition* value.

The approach's complexity can be reduced by introducing a buffer to store the already queried function values and thus omitting some of the costly function calls, as the investigated areas often overlap. This is evaluated in the experiment Section 6.2. Furthermore, when working on large databases, the method can be applied to a smaller, however still representative subset of the database, which further reduces the number of required function calls.

The proposed solution comes with a drawback: it requires a number of parameters to be set, which is contrary to our stated goal of a parameter free annotator. But during our experiments in Section 6.2 we show that the parameter optimization procedure is fairly robust to its own initial parameters and that the same is not true for PATSI with respect to the selection of t and k. The iterative refinement parameters can therefore be fixed independently from the PATSI configuration.

6 Experimental study of the PATSI method

The experiments were planed with a number of research questions in mind and seek to investigate our method's most important characteristics.

Deciding on the most suitable transfer function is crucial for the annotator performance. The first Section 6.1 is therefore dedicated to experiments testing the performance and attributes of the eight transfer functions proposed in Section 3.2.3. Beside our main goal of automatically deciding the annotation length, we also strive for PATSI to be as simple, automatized and efficient as possible. In Section 5 we introduced an iterative refinement algorithm to obtain semi-optimal values for the PATSI parameters t and k. This method is computational expensive and depends itself on a number of parameters. In Section 6.2 we therefore verify the need for optimizing the neighbourhood k and the threshold t and investigate the algorithm's robustness to its initial parameters. Furthermore we plan and execute experiments investigating the performance of various similarity spaces in conjunction with PATSI. The settings and results of this exhaustive feature and distance measure comparison can be found in Section 6.3. Such prepared with a good choice of the TF, the SS and parameters optimized by the iterative refinement procedure, we test PATSI on a number of popular benchmark databases and compare its performance against a number of state-of-the-art automatic image annotation algorithms in Section 6.4.

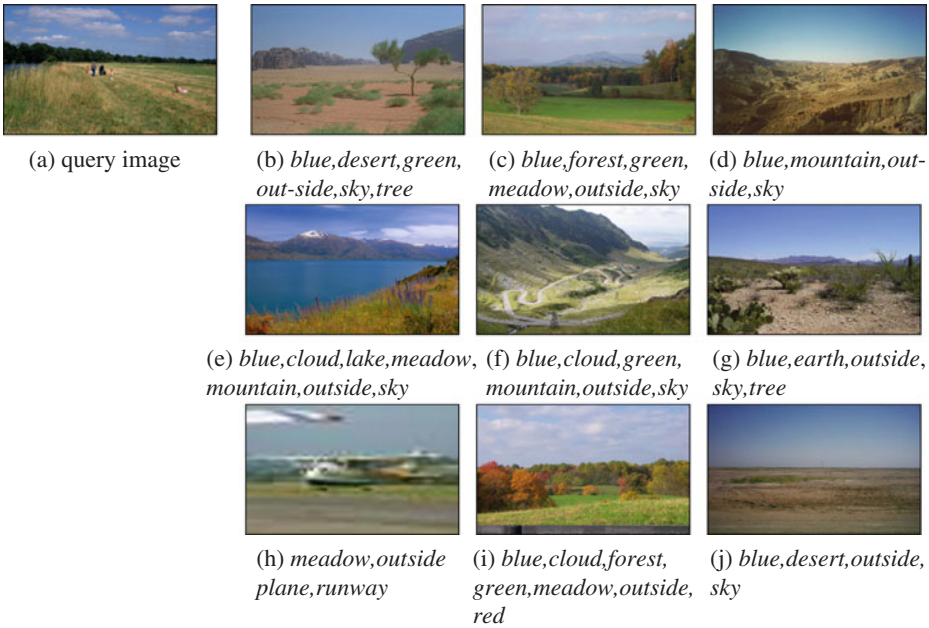
If not otherwise stated, all experiments were performed with the same evaluation framework: To evaluate the resulting annotations' quality, the three evaluation met-

Table 1 Properties of MGV 2006 training dataset

	MGV 2006
Number of images	751
Dictionary size	74
Mean annotation length	5.0
Mediana of annotation length	5.0
Std. dev. of annotation length	1.28
Min. and max. annotation length	(2, 9)

rics precision, recall and F-score were employed, as they are described in Section 6. To obtain expressive results, 4-fold cross validation is applied and the average F-score value is calculated. Generally the experiments were conducted using the MGV dataset (Paradowski 2008; Kwasnicka and Paradowski 2008), whose details can be found in Table 1. It is of acceptable size to allow for fast but still representative experiment execution. Beside the low computational complexity, it shows good results and is well annotated.

A result obtained with PATSI on this dataset are presented in Fig. 7. In the current version of PATSI, the query image (Fig. 7a) gets the words *blue, cloud, green, meadow, outside, sky, tree, desert, forest, mountain* assigned. If image f would be rightly annotated with *meadow* and image c with *cloud*, the threshold t could be raised to exclude *forest* and *mountain*. Using a taxonomy to exploit the association between *forest* with *tree* would furthermore allow the exclusion of *desert*, leading to a ideal annotation.



(a) query image

(b) *blue,desert,green, out-side,sky,tree*

(c) *blue,forest,green, meadow,outside,sky*

(d) *blue,mountain,out-side,sky*

(e) *blue,cloud,lake,meadow, mountain,outside,sky*

(f) *blue,cloud,green, mountain,outside,sky*

(g) *blue,earth,outside, sky,tree*

(h) *meadow,outside plane,runway*

(i) *blue,cloud,forest, green,meadow,outside, red*

(j) *blue,desert,outside, sky*

Fig. 7 Example run of PATSI with query image (**a**) and its nine nearest neighbours (**b**–**g**). The annotations *blue, cloud, green, meadow, outside, sky, tree* form the ground-truth of the query image

6.1 Comparison of the proposed transfer functions

In this section we investigate the transfer function's influence on the annotation quality. We test the TFs defined in Section 3.2.3, which are all shaped such that more similar images exercise a stronger influence the resulting annotation. The experiments are performed on the MGV image dataset, CEDD feature set (Chatzichristofis and Boutalis 2008a) and Minkowski L2 distance measure. Figures 8 and 9 show the variations in the resulting F-score for each of the TF for different k and t combinations.

The peaks are of similar height i.e., the highest achievable qualities do not differ greatly between the TFs. The only exception is the e^{-d} transfer function's performance, the evaluation of e^{-d^2} is subsequently stopped. But the global maxima's position differs greatly between the functions. We therefore apply the proposed iterative optimization method to find good values for $k*$ and $t*$. Next, we used these values to train the PATSI annotator. To obtain more representative results, we applied leave-one-out instead of 4-fold cross-validation approach during this experiment. For a simple comparison such as this, the type of cross-validation is of no relevance. In Table 2 we present the $k*$ and $t*$ obtained by the optimization method and the resulting F-score for each TF.

The distance based TFs perform overall poorer than the rank based TFs. This could be because they strongly rely on the distance measure employed and a single distance measure is unlikely to capture more than a subset of semantic concepts. One can suppose that assuming an 'ideal' distance measure, the distance based would outperform the rank based TFs. For now we recommend rank based TFs, as they stress the number of similar images rather than their actual distance from the query

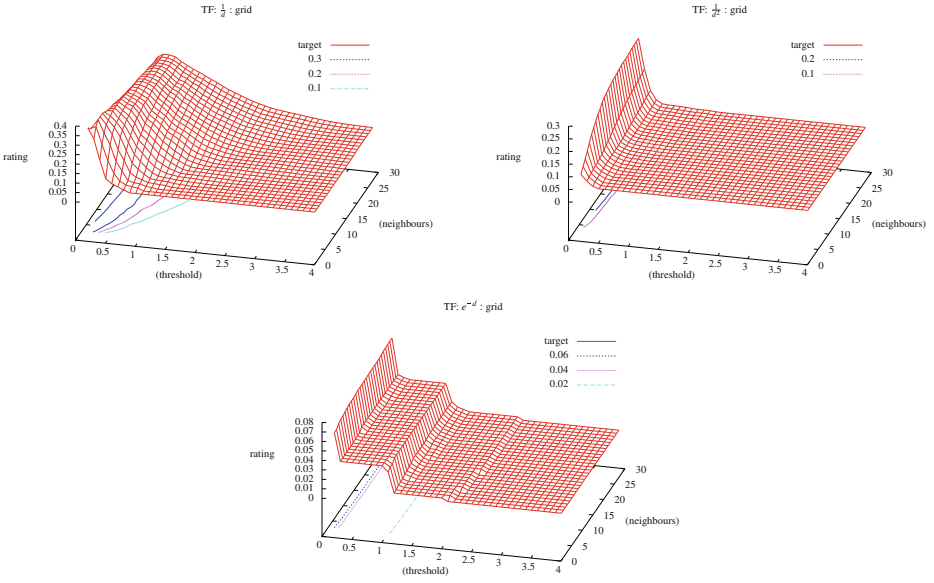


Fig. 8 F-score of annotations obtained with different transfer functions (distance based transfer functions given by (5)–(7)), MGV data set, CEDD feature set, and Minkowski L2 distance measure

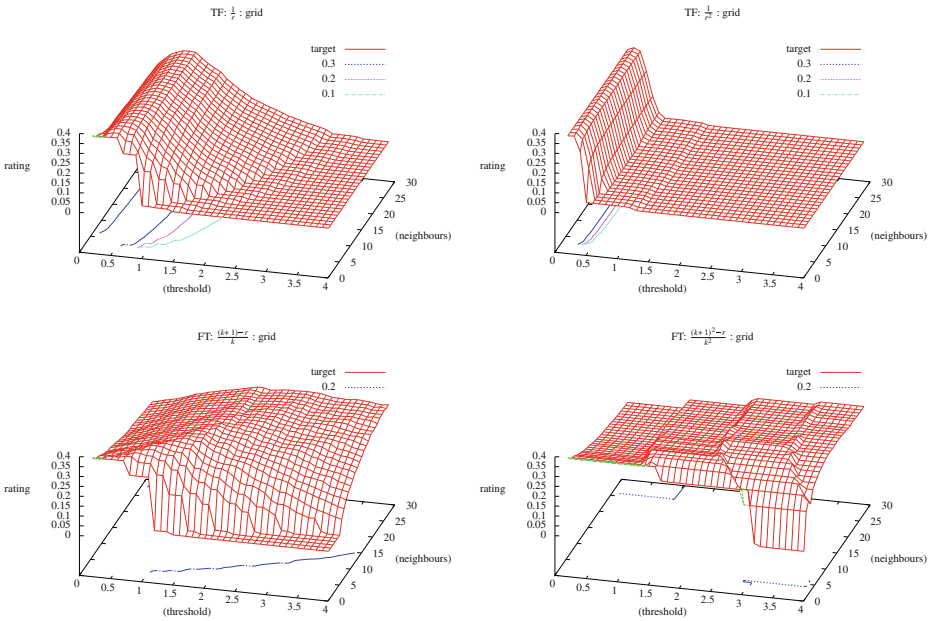


Fig. 9 F-score of annotations obtained with different transfer functions (rank based transfer functions given by (9)–(12)), MGV data set, CEDD feature set, and Minkowski L2 distance measure

image, and are therefore able to smooth out errors induced by the distance measures. This assumption is supported by the experimental results, placing the rank based above the distance based TFs.

Comparing the four proposed TFs from the rank based group, we can observe that their performance does not differ greatly. We chose $1/r$ for its slightly better results, simplicity and the greatest number of similar images taken into account.

6.2 Evaluation of the optimization procedure

The optimization procedure's target is to yield semi-optimal values for t and k. We treat the annotators quality as measured by its F-score as a function of t and k whose

Table 2 Evaluation of transfer functions

Transfer function	$k*$	$t*$	F-score
Distance based			
$\frac{1}{d}$	7	0.18	0.35804
$\frac{1}{d^2}$	23	0.06	0.32253
e^{-d}	94	0.1	0.07238
Rank based			
$\frac{1}{r}$	19	0.73	0.38107
$\frac{1}{r^2}$	7	0.15	0.35877
$\frac{(k+1)-r}{k}$	10	1.3	0.37517
$\frac{(k+1)^2-r}{k^2}$	7	2.41	0.3732

d = distance, r = rank, k = neighbourhood size

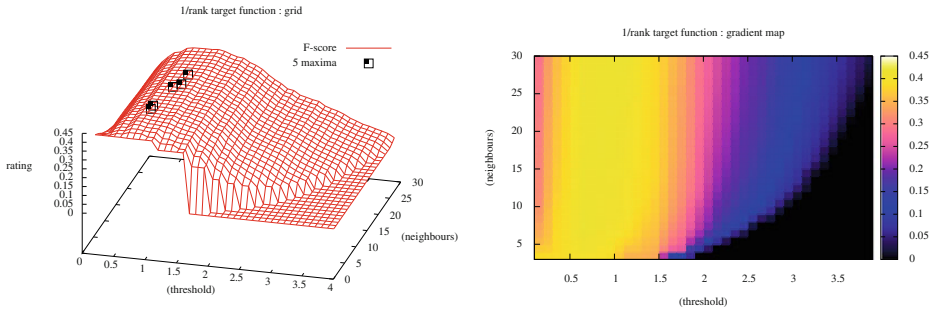
global maxima we seek. The shape of the function to be optimized is presented in Fig. 10. Figure 10a presents the 3D shape of F-score as a function of k and t, where the five highest points are marked by small crosses. Figure 10b shows the same function as its gradient (a bird's eye perspective).

Observing this figure and the graphs presented in Figs. 8 and 9, it can be seen that although the function appears linear, it is in fact non-linear and, due to the discrete domain of the neighbourhood size k, half-discrete and half-continuous. The functions show some deep descends, so that even small changes in t and/or k can lead to sudden drops in the annotators performance. Such the values for t and k strongly influence the F-score and are difficult to set manually.

Therefore the use of our proposed optimization method is justified. In this section we perform some experiments to show the methods independence from its parameters. Furthermore we compare the performance of different approaches.

During the execution of the iterative refinement an already queried point is often requested again in a later iteration step, so that using a buffer to store the calculated performed annotations (performing annotation requires a lot of time) can reduce the computational complexity. In addition, for the k lie close to the assumed range, it is impossible to find a sufficient number of neighbours to get the given threshold (t set at that point). To verify the above point of view, in our experiments we measure the number of function calls (responsible for computation complexity) with no buffering, with buffer, and with buffering combined with omitting some k values (near the border of the search space). Therefore the results of the experiments are collected as a *real*, *overall*, and *cleaned*. *Overall* is a number of annotation calls according to the proposed algorithm (without using any buffers). *Real* is a number of the annotations with caching calculations (for every pairs t and k the annotation is calculated only once). *Cleaned* is a number of annotations using caching and omitting such k for which too small number of neighbours exist (k is close to the border and the sum of transfers cannot exceed a given t).

The main question to the experiments is: does a fixed set of optimization procedure parameters is suitable for different dataset, distance measure and feature set combinations? In the previous section we have decided to use $\frac{1}{rank}$ transfer function so this function was employed together with Minkowski L2 distance measure. Two



(a) F-score values on a grid (b) gradient representation of the same F-score

Fig. 10 F-score of annotation in the optimization procedure (MGV2006, CEDD, Minkowski L2)

feature sets were used in experiments: CEDD (Chatzichristofis and Boutalis 2008a) and FCTH (Chatzichristofis and Boutalis 2008b).

6.2.1 Sensitivity of the optimization procedure on the initial area of the search space

We defined three scenarios:

Area I: Starting with a very small initial area (the optimization procedure is forced to increase the bounds of searching).
Area II: Start with a huge area (it strongly increases the amount of annotation transfers, that have to be calculated).
Area III: Tune the search on a small area where the highest F-scores are expected.

These settings let to the results displayed in Table 3. Investigating a very wide area, as it is done in II, is computationally clearly inefficient. The resulting quality is even lower than with the tuned area (case III), while a number of required annotations is roughly five times higher. Starting with a very limited area, as in I, reduces the number of required annotation transfers significantly. For the given examples the results are also acceptable regarding F-score measure. But firstly, the low number of neighbours means that only very few similar images are considered for selecting their annotation to transfer. This can lead to a low recall, violating the target of a balanced recall and precision measures. Furthermore, the high quality cannot be assured for other annotator configurations, as the shape of the target function will be the same, but the position of the plateau of the highest values might differ. Starting with a sufficiently big area (case III) is the most suitable approach and can be expected to lead to good results for all possible annotator configurations.

6.2.2 Sensitivity of the optimization procedure on the divider

Influence of the divider on the optimization procedure was tested using two scenarios.

Divider I: A low divider (only a slightly smaller area is considered in each iteration step).
Divider II: A higher divider (a considered area decreasing more rapidly in each iteration step).

Table 3 Varied initial areas for the optimization algorithm on MGV 2006 dataset

Initial area			Performance in queries			Results		
	Threshold	Neighbours	Cleaned	Real	Overall	F-score	t^*	k^*
CEDD								
I	[0.99, 1.01]	[1, 1]	40	50	55	0.34588	0.35	3
II	[0.1, 10]	[2, 50]	1188	1786	1921	0.36014	0.745	19
III	[0.7, 1]	[5, 23]	209	209	301	0.36599	0.633	10
FCTH								
I	[0.99, 1.01]	[1, 1]	65	81	94	0.33439	0.43	4
II	[0.1, 10]	[2, 50]	1170	1768	1869	0.34465	0.643	20
III	[0.7, 1]	[5, 23]	236	236	343	0.34151	0.541	24

Table 4 Varied initial dividers for the optimization algorithm on MGV 2006 dataset

Initial divider		Performance in queries			Results		
		Cleaned	Real	Overall	F-score	t^*	k^*
CEDD							
I	2	299	299	434	0.36014	0.745	19
II	3	414	414	481	0.36116	0.528	19
FCTH							
I	2	287	287	399	0.34465	0.643	20
II	3	324	314	358	0.34237	0.666	28

Received results are presented in Table 4. The resulting qualities using different initial dividers are similar in F-score, $k*$ and $t*$. The computational complexity is slightly lower for I, therefore it is the setting of choice.

6.2.3 Sensitivity of the optimization procedure on the initial threshold value

Here we defined three scenarios.

Initial threshold resolution I: A very low initial threshold (a finer scanning of the area in each iteration step).

Initial threshold resolution II: A medium initial threshold (a balance between the costs and efficiency of the scanning).

Initial threshold resolution III: A high initial threshold (only broad scanning of the area in each iteration step).

The results are collected in Table 5. The resulting qualities using different initial thresholds are similar in F-score, $k*$ and $t*$. As we expected, a broader scan (case III) leads to a reduced number of necessary annotations. With case II, the best value found takes into account a few more neighbours than in III, while the gain in computational efficiency of III compared with II is only marginal. Therefore case II is the setting chosen.

It can be noted that the settings of the optimization procedure parameters, while sometimes strongly influencing the performance, seem to have only a low impact on the resulting quality caused by receiving better $k*$ and $t*$ values. It can be assumed that the optimization procedure with the selected moderate settings will lead to good

Table 5 Varied initial thresholds for the optimization algorithm on MGV 2006 dataset

Initial threshold		Performance in queries			Results		
		Cleaned	Real	Overall	F-score	t^*	k^*
CEDD							
I	0.1	600	601	650	0.36511	0.531	13
II	0.64	299	299	434	0.36014	0.745	19
III	1.0	149	164	257	0.36663	0.506	7
FCTH							
I	0.1	617	618	669	0.34888	0.578	11
II	0.64	287	287	399	0.34465	0.643	20
III	1.0	183	198	308	0.34888	0.579	11

results for all annotator configurations using $\frac{1}{rank}$ as transfer function. This comes together with a relatively low computational complexity. We can say that our target of obtaining initial parameters for the optimization method that are invariant to the annotators configuration has been reached. PATSI can therefore be treated as a parameter free method.

6.3 PATSI results with different similarity spaces

Recall that the term Similarity Space (SS) denotes a combination of a distance measure and a feature set. Although research on CBIR has been performed since years, the features most suitable for the task are still unknown (Deselaers et al. 2008b). In this section we present a number of distance measures and feature sets to investigate their performance with PATSI in all possible combinations. The experiments were conducted on the MGV dataset with 4-fold cross-evaluation.

The following six types of features were considered:

1. From **MPEG-7** standard Chang et al. (2001) we use following image descriptors calculated for the whole image:
   - Fuzzy Color Histogram—125 dimensions,
   - JPEG Coefficient Histogram—192 dimensions,
   - General Color Layout—18 561 dimensions,
   - Color and Edge Directivity Descriptor (CEDD)Chatzichristofis and Boutalis (2008a)—120 dimensions,
   - Fuzzy color and texture histogram (FCTH)Chatzichristofis and Boutalis (2008b)—192 dimensions.
2. **Tamura features**—first three from six texture features corresponding to human visual perception (Tamura et al. 1978):
   - coarseness—size of the texture elements,
   - contrast—contrast stands for picture quality,
   - directionality—texture orientation.

   Tamura features is 16-dimensional vector.
3. **Auto Color Correlogram** features defined in (Goodrum 2000; Huang et al. 1997)—256 dimensions
4. **Gabor** texture features (Zhang et al. 2000)—60 dimensions
5. Statistical colour and edges information of image regions (5-by-5 and 20-by-20 grid) in two colour spaces RGB and HSV:
   - x and y coordinates of the segment centre—2 dimensions,
   - the mean value of colour in each channel of the colour space—3 dimensions,
   - standard deviations of colour changes in each channel for a given colour space—3 dimensions,
   - mean eigenvalues of colour Hessian in each channel for a given colour space—3 dimensions.
6. **CoOccurance Matrix** (Haralick et al. 1973) calculated for each segment of 5-by-5 and 20-by-20 segmentation—21 dimensions.

Table 6 Evaluation of distance measure and local feature set combinations

Distance	Features	Precision	Recall	F-score
Cosine	hsv/dev	0.2862725	0.3777475	0.32543
	CoOccuranceMatrix	0.1869675	0.2736875	0.222115
	xy/rgb/dev/hes	0.2887575	0.320405	0.302915
	xy/hsv/dev/hes	0.2935	0.36891	0.326575
	hsv/dev/hes	0.28504	0.3717225	0.3224325
	rgb	0.2537375	0.302795	0.275145
	rgb/dev/hes	0.2670425	0.3495275	0.3009725
	hsv	0.272935	0.305805	0.28786
	rgb/dev	0.2637025	0.3467275	0.29778
Minowskil2	hsv/dev	0.256685	0.355055	0.296665
	CoOccuranceMatrix	0.1973575	0.26986	0.2231
	xy/rgb/dev/hes	0.2896725	0.29585	0.292305
	xy/hsv/dev/hes	0.257225	0.35584	0.297305
	hsv/dev/hes	0.257225	0.35584	0.297305
	rgb	0.2643425	0.285675	0.274505
	rgb/dev/hes	0.2896725	0.29585	0.292305
	hsv	0.2485225	0.3456775	0.28878
	rgb/dev	0.283145	0.300085	0.29075
Minowskil1	hsv/dev	0.342625	0.37346	0.3570425
	CoOccuranceMatrix	0.190515	0.299885	0.2322825
	xy/rgb/dev/hes	0.30654	0.306565	0.3045675
	xy/hsv/dev/hes	*0.337825*	*0.38379*	*0.359095*
	hsv/dev/hes	0.34391	0.3819575	0.36166
	rgb	0.2587375	0.33588	0.2917025
	rgb/dev/hes	0.2999625	0.316095	0.3061425
	hsv	0.30599	0.3791275	0.3368775
	rgb/dev	0.311385	0.313525	0.3121075
Correlation	hsv/dev	0.2610375	0.350125	0.298755
	CoOccuranceMatrix	0.1852525	0.264855	0.2178775
	xy/rgb/dev/hes	0.27289	0.3263425	0.2955575
	xy/hsv/dev/hes	0.2819825	0.34359	0.308305
	hsv/dev/hes	0.2700475	0.34293	0.3017975
	rgb	0.268855	0.2915275	0.2775225
	rgb/dev/hes	0.2836325	0.3165125	0.29856
	hsv	0.2483675	0.3366125	0.283445
	rgb/dev	0.2682975	0.322935	0.291995
kl	CoOccuranceMatrix	0.05934	0.25594	0.0951
	rgb	0.136465	0.3539925	0.195195
	rgb/dev/hes	0.1310725	0.3807725	0.1926425
	hsv/dev/hes	0.1823825	0.298395	0.22593
	rgb/dev	0.1357975	0.3683475	0.19679
js	CoOccuranceMatrix	0.0576525	0.2781	0.0952825
	rgb	0.1369	0.44401	0.20411
	rgb/dev/hes	0.1682025	0.3455425	0.220285
	hsv/dev/hes	0.1693175	0.3231575	0.221815
	rgb/dev	0.1484025	0.409405	0.21037

For the experiments we differ between global features (take from the Lire Package by Lux and Chatzichristofis (2008))

1. Auto Color Correlogram
2. CEDD
3. FCTH
4. Fuzzy Color Histogram
5. General Color Layout
6. Jpeg Coefficient Histogram
7. Gabor
8. Tamura

Table 7 Evaluation of distance measure and global feature set combinations

Distance	Features	Precision	Recall	F-score
Cosine	cedd	0.3289175	0.35534	0.341555
	GeneralColorLayout	0.151765	0.3024625	0.1977775
	Gabor	0.0899975	0.25198	0.1310075
	Tamura	0.1734525	0.27898	0.2121925
	JpegCoefficientHistogram	0.2486975	0.3395125	0.282435
	AutoColorCorrelogram	0.212885	0.31262	0.25317
	fcth	0.282625	0.40072	0.33133
	fuzzyColorHistogram	0.20783	0.271075	0.2324675
Minowskil2	cedd	0.329165	0.36847	0.347475
	GeneralColorLayout	0.1553375	0.2808125	0.1995
	Gabor	0.0935425	0.2586125	0.1369275
	Tamura	0.176785	0.32527	0.2288475
	JpegCoefficientHistogram	0.2549675	0.31469	0.280865
	AutoColorCorrelogram	0.20977	0.3400225	0.2586875
	fcth	0.3037425	0.364405	0.330335
	fuzzyColorHistogram	0.2031825	0.300225	0.24129
Minowskil1	cedd	0.3161275	0.3662125	0.33805
	GeneralColorLayout	0.159835	0.3580725	0.2205675
	Gabor	0.0976975	0.2820875	0.14506
	Tamura	0.18575	0.2981875	0.22849
	JpegCoefficientHistogram	0.2430325	0.36182	0.2905975
	AutoColorCorrelogram	0.230605	0.2916075	0.25615
	fcth	0.29227	0.3994575	0.337125
	fuzzyColorHistogram	0.1778075	0.304465	0.224205
Correlation	cedd	0.3228425	0.362165	0.3411125
	GeneralColorLayout	0.1423775	0.2994125	0.19219
	Gabor	0.089965	0.25981	0.131615
	Tamura	0.1939625	0.258	0.2209275
	JpegCoefficientHistogram	0.2475275	0.334825	0.2824825
	AutoColorCorrelogram	0.2137975	0.310405	0.252855
	fcth	0.29422	0.3952575	0.3372625
	fuzzyColorHistogram	0.22045	0.260525	0.238385

and a number of local features in different combinations that were extracted from a grid:

1. colour means in RGB colour space
2. colour means in RGB + colour deviations
3. colour means in RGB + dev. + mean eigen value in colour hessian
4. colour means in RGB + dev. + mean eigen value in colour hessian with x/y coordinates
5. colour means in HSV colour space
6. colour means in HSV + colour deviations
7. colour means in HSV + colour deviation + mean eigen value in colour hessian
8. colour means in HSV + colour deviation + mean eigen value in colour hessian with x/y coordinates
9. Cooccurance Matrix (Haralick et al. 1973)

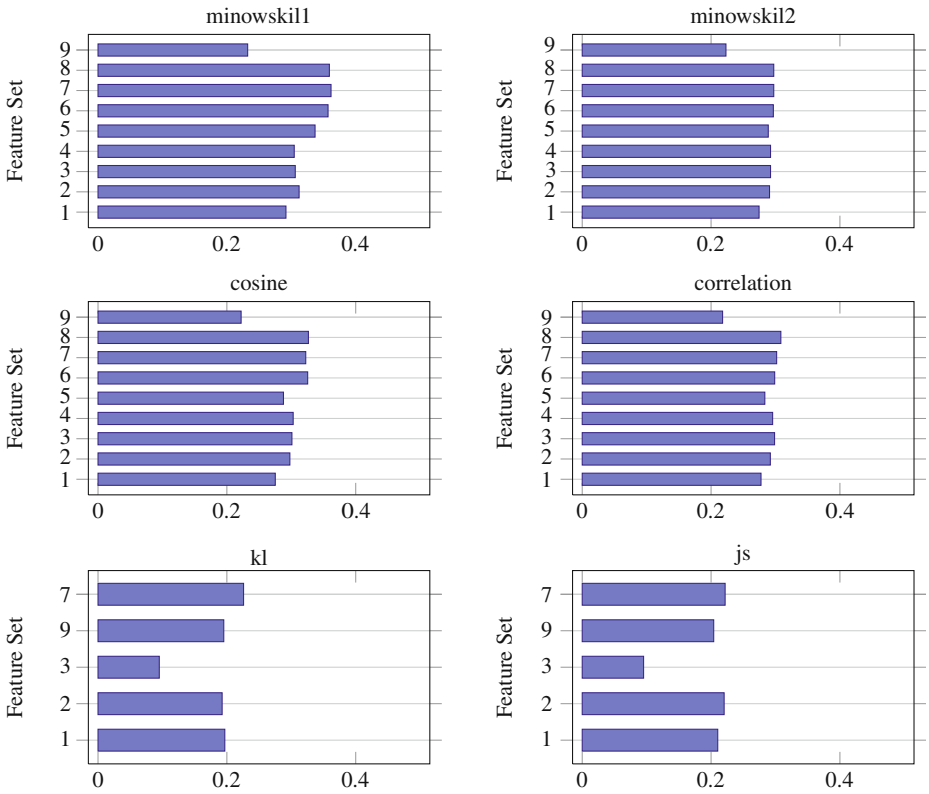


Fig. 11 F-score of distance measures with local features grouped by distance measures. The number refers to the corresponding local feature set in the listing at the beginning of Section 6.3

To obtain a SS these were combined with a number of distance measures, namely:

– Metric Space:

1. Minowski L1,
2. Minowski L2,
3. Correlation,
4. Cosine.

– Probabilistic Space:

5. Kullback Leibler (one directional),
6. Jehnsen Shannon (two directional).

Note that Kullback Leibler (KL) and Jehnsen Shannon (JS) were only applied to some local and no global features. All together a number of 79 SSs were tested. Their evaluations are shown in Table 6 for local and Table 7 for global features. The highest value of all is written in italics.

For better perception and easier comparison, the results are also presented in two sets of bar graphs, once grouped by distance measure (Figs. 11 and 12), and once by feature sets (Figs. 13 and 14).

The quality achieved through the SSs differs greatly over a range from 0.095 for KL with CoOccuranceMatrix up to 0.359 for Minowski L1 with xy/rgb/dev/hes feature set. This shows that the choice of distance measure and feature set is crucial. But it does not suffice to simply choose the best distance measure and combine it

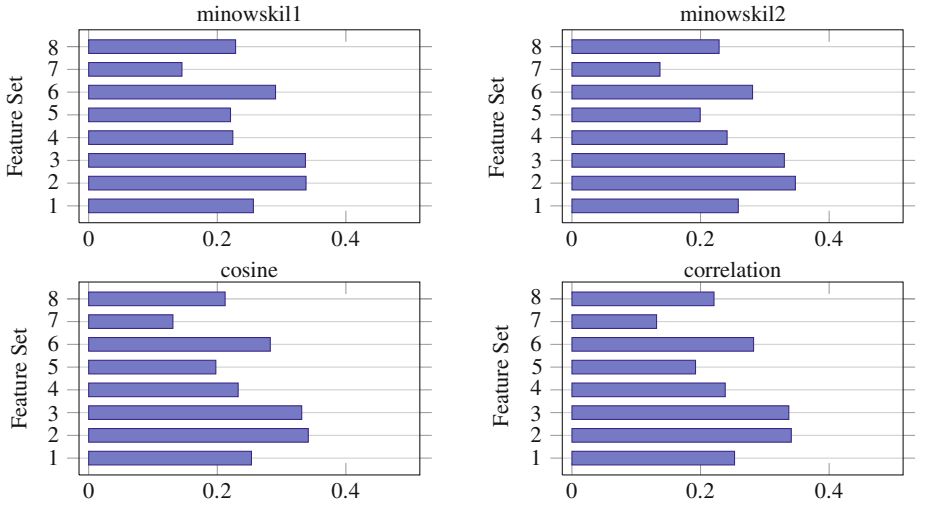


Fig. 12 F-score of distance measures with global features grouped by distance measures. The number refers to the corresponding global feature set in the listing at the beginning of Section 6.3

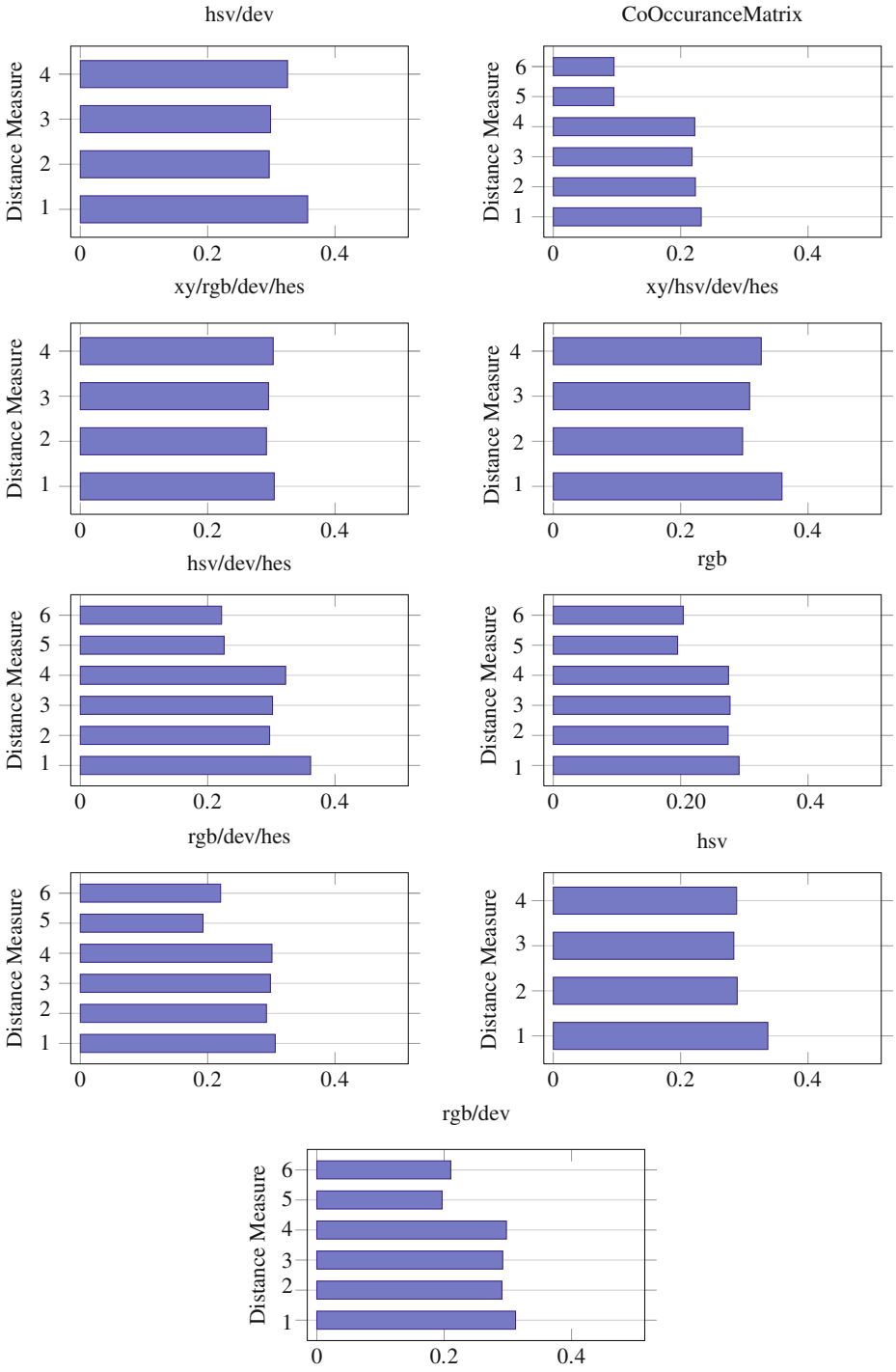


Fig. 13 F-score of distance measures with local features grouped by feature sets. The number refers to the corresponding distance measure in the listing at the beginning of Section 6.3

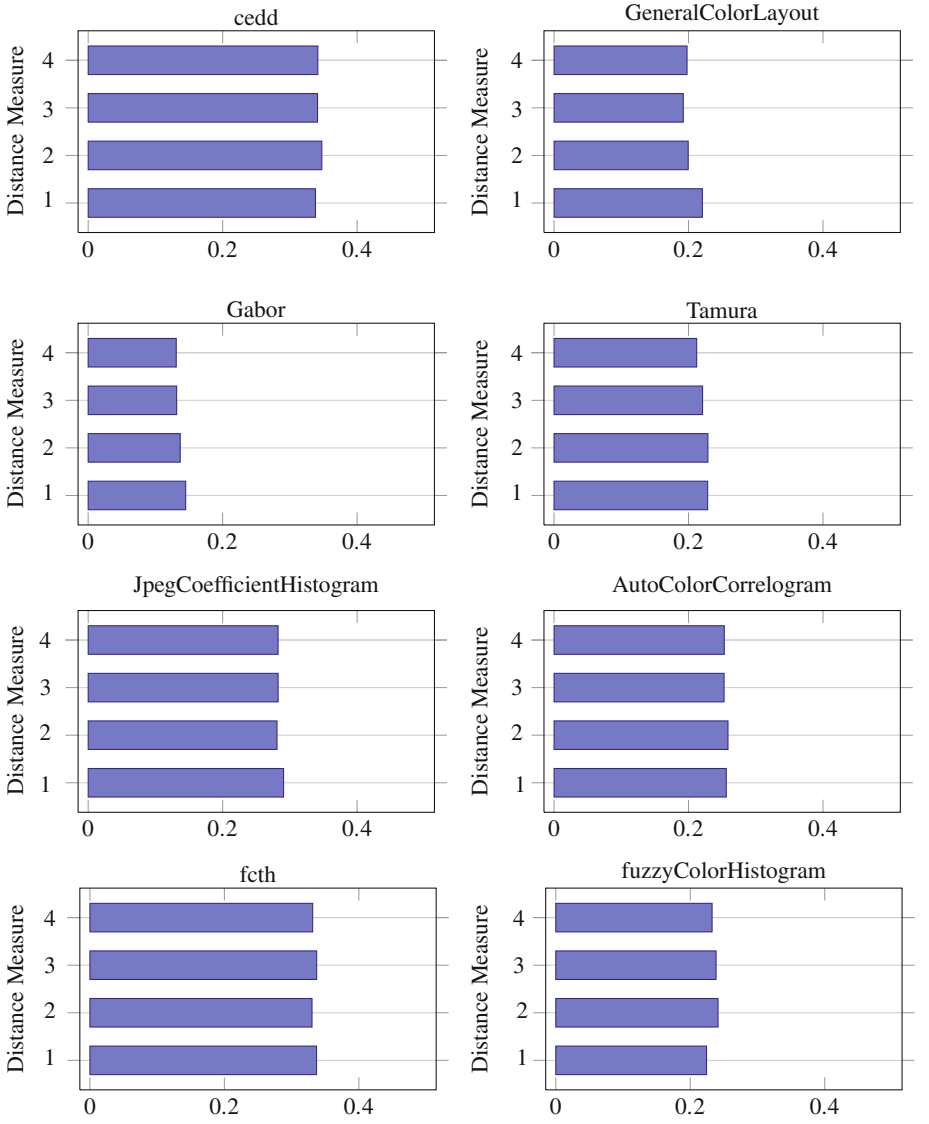


Fig. 14 F-score of distance measures with global features grouped by feature sets. The number refers to the corresponding distance measure in the listing at the beginning of Section 6.3

with the best feature set: the bar-graphs show that the performance of a distance measure varies strongly with the feature set on which it was applied. This is especially observable with the global features (Fig. 12). A performance of different feature sets does not vary so much with the applied distance measure, with the notable exception of KL and JS. This robustness is the most dominant with the global features as can be observed in Fig. 14, where the choice of distance measure has nearly no impact on the quality.

Table 8 Properties of benchmark datasets

	MGV 2006	ICPR 2004	IAPR TC-12
Number of images	751	1109	19805
Dictionary size	74	407	291
Mean annotation length	5.0	5.79	5.72
Mediana of annotation length	5.0	5.0	5.0
Std. dev. of annotation length	1.28	3.48	2.56
Min. and max. annotation length	(2, 9)	(1,23)	(1,23)

Table 9 Evaluation of image annotation algorithms on MGV 2006 dataset

Method	Precision	Recall	F-score
FastDIM	0.24	0.16	0.19
FastDIM + GRWCO	0.34	0.34	0.34
MCML	0.32	0.24	0.27
MCML + GRWCO	0.38	0.37	0.37
CRM	0.39	0.34	0.36
PATSI	**0.38**	**0.46**	**0.42**
The best 20 words			
FastDIM	0.58	0.53	0.51
FastDIM + GRWCO	0.59	0.61	0.60
MCML	0.61	0.59	0.60
MCML + GRWCO	0.64	0.62	0.63
CRM	0.58	0.57	0.57
PATSI	**0.71**	**0.86**	**0.78**

The bold entries are the results of our method (PATSI), compared to other methods from literature (not in bold)

Table 10 Evaluation of image annotation algorithms on ICPR 2004 dataset

Method	Precision	Recall	F-score
FastDIM	0.20	0.17	0.18
FastDIM + GRWCO	0.21	0.21	0.21
MCML	0.21	0.17	0.19
MCML + GRWCO	0.25	0.28	0.26
CRM	0.24	0.24	0.24
PATSI	**0.27**	**0.34**	**0.30**
The best 60 words			
FastDIM	0.64	0.58	0.61
FastDIM + GRWCO	0.63	0.61	0.62
MCML	0.69	0.60	0.64
MCML + GRWCO	0.69	0.67	0.68
CRM	0.61	0.61	0.61
PATSI	**0.82**	**0.94**	**0.88**

The bold entries are the results of our method (PATSI), compared to other methods from literature (not in bold)

Table 11 Evaluation of image annotation algorithms on IAPR TC 12 dataset

Method	Precision	Recall	F-score
RGB	0.24	0.24	0.24
HSV	0.20	0.20	0.20
LAB	0.24	0.25	0.24
Haar	0.20	0.11	0.14
HaarQ	0.19	0.16	0.17
Gabor	0.15	0.15	0.15
GaborQ	0.08	0.09	0.08
MBRM	0.24	0.23	0.23
Lasso	0.28	0.29	0.28
JEC	0.28	0.29	0.28
PATSI	**0.26**	**0.31**	**0.28**

The bold entries are the results of our method (PATSI), compared to other methods from literature (not in bold)

6.4 PATSI quality compared to similar methods in literature

For the evaluation process three benchmarking data sets were used: ICPR 2004 (ICPR 2004), MGV 2006 (Paradowski 2008) and IAPR TC-12 (Grubinger et al. 2006), whose characteristics are shown in Table 8.

For the MGV and ICPR datasets as the reference points we have obtained the results presented in Kwasnicka and Paradowski (2008). For these data sets the proposed method achieved significantly better results. The highest difference is seen for the best annotated words, where F-score was improved by 20 percentage points in both sets (the relative improvement over the CRM method is about 37% and 44% for the MGV and ICPR datasets respectively) (Tables 9 and 10).

For IAPR TC 12 as the reference point we have acquired the results presented in Makadia et al. 2008. We obtained comparable results to the Lasso and JEC methods on that benchmark set. Lasso and JEC equally used the approach of transferring annotation from similar images, but both of these methods combine seven different similarity measures and feature sets, such as RGB, HSV, LAB, Haar, HaarQ, Gabor and GaborQ. Only a combination of those measures would allow for comparable results to PATSI. The method proposed by Makadia et al. (2008) cannot automatically determine the annotation's length, assuming that this is one of the given parameters (Table 11).

7 Conclusion

The performed experiments show that the proposed PATSI method fulfills the assumed requirements, i.e. it produces good results using single similarity measure (in contrast to the runner-ups Lasso and JEC from Makadia et al. (2008)) and does not require manually tuning its parameters. It is worth to underline that the PATSI method produces annotations with variable length, as it is the most important novelty that distinguishes PATSI from other approaches.

Every image is a collection of visual information which can be annotated by a different number of words, because visual information can be stored in parallel. Using automatic image annotation with a fixed annotation length does not cope with this varying number of concepts, leading to unrealistic and limited annotations. Using variable annotation length better reflects the reality.

Most approaches towards automatic image annotation depend on a number of parameters which must be tuned manually, often by trial and error method. This makes them unsuitable for usage on a fast investigation into the space of other configurations as different distance measures and feature sets. PATSI with the iterative refinement method allows to automatically determine good values for the annotator's parameters.

Despite its good performance PATSI also shows some weaknesses. Our method's complexity depends on the annotator knowledge, as the query image is compared to each image of the training database. This makes PATSI unsuitable for very large image collections.

A second shortcoming is the closed vocabulary of PATSI. Unknown semantic concepts cannot be annotated and every change in the vocabulary requires a re-execution of the computationally expensive learning phase. This renders the method unsuitable for fast-changing web applications.

Furthermore PATSI does not consider the vocabulary taxonomy. If, for example, a query image's nearest neighbours contain the words *man* and *men*, our annotation transfer treats them as independent. A situation can arise, where their individual weights are not sufficient to pass the threshold t, while a consideration of their combined weight would lead to a correct transfer.

During our experiments we observed that some SSs were good at detecting and correctly transferring some groups of keywords, while others performed better for other groups. Different features capture different aspects of the human perception of similarities between images. We believe that a real increase in the performance of automatic image annotation is possible using a combination of SSs, where each SS is responsible for a group of keywords belonging to a common similarity concept. The presented iterative refinement procedure together with the undertaken evaluation of a number of SSs allow for a fast investigation into their individual performances for certain semantic groups of keywords, and to select a suitable set of SSs for a combined approach.

Furthermore it would be interesting to study the impact of annotation word correlation, as just recently proposed by Zhang et al. (2011). Their method exploits correlations between the tags and the images' visual features, making tag probability information available that can be utilized during the annotation process. This promising method could be easily integrated with PATSI.

To decrease the computational complexity of our method, animate vision could be employed (Boccignone et al. 2008) which would allow to concentrate only on the image regions with the highest probability of relevance. To reduce the search space, which can be very large for huge image databases, Wichert (2008) proposed a hierarchical linear subspace method. Adapting this approach to other than colour-based features could lead to an significant speed up for PATSI.

To tackle the problem of relationships between words, one of the taxonomies described in Tousch et al. (2012) like WordNet could be employed together with an extended evaluation measure such as proposed in Nowak et al. (2011).

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

Aoun, N. B., Elghazel, H., Hacid, M. S., & Amar, C. B. (2011). Graph aggregation based image modeling and indexing for video annotation. In: *Proceedings of the 14th international conference on computer analysis of images and patterns, CAIP'11* (Vol. Part II, pp. 324–331). Springer-Verlag, Berlin, Heidelberg.

Boccignone, G., Chianese, A., Moscato, V., & Picariello, A. (2008). Context-sensitive queries for image retrieval in digital libraries. *Journal of Intelligent Information Systems*, *31*, 53–84.

Carneiro, G., Chan, A., Moreno, P., & Vasconcelos, N. (2007). Supervised learning of semantic classes for image annotation and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(3), 394–410.

Carson, C., Belongie, S., Greenspan, H., & Malik, J. (2002). Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 1026–1038.

Chang, S. F., Sikora, T., & Puri, A. (2001). Overview of the MPEG-7 standard. *IEEE Transactions on Circuits and Systems for Video Technology*, *11*(6), 688–695.

Chang, E., Goh, K., Sychay, G., & Wu, G. (2003). Cbsa: Content-based soft annotation for multimodal image retrieval using bayes point machines. *IEEE Transactions on Circuits and Systems for Video Technology*, *13*(1), 26–38.

Chatzichristofis, S. A., & Boutalis Y. S. (2008a). CEDD: Color and Edge Directivity Descriptor: A compact descriptor for image indexing and retrieval. In: *Proceedings of the 6th Int. Conf. on Computer Vision Systems (IVCS '08)* (pp. 312–322).

Chatzichristofis, S. A., & Boutalis, Y. S. (2008b). Fcth: Fuzzy color and texture histogram—A low level feature for accurate image retrieval. In: *International workshop on image analysis for multimedia interactive services* (Vol. 0, pp. 191–196).

Cusano, C., Ciocca, G., & Schettini, R. (2004). Image annotation using svm. In: *Proceedings of SPIE* (Vol. 5304, pp 330–338).

Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, *40*, 5:1–5:60.

Deselaers, T., Keysers, D., & Ney, H. (2008a). Features for image retrieval: An experimental comparison. *Information Retrieval*, *11*, 77–107. doi:10.1007/s10791-007-9039-3, http://dl.acm.org/citation.cfm?id=1349658.1349663.

Deselaers, T., Keysers, D., & Ney, H. (2008b). Features for image retrieval: An experimental comparison. *Information Retrieval, 11*(2), 77–107.

Duygulu, P., Barnard, K., de Freitas, J. F. G., & Forsyth, D. A. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: *Proc. of the 7th European conf. on computer vision, Springer, London, UK*.

Eakins, J. P., Briggs, P., & Burford, B. (2004). Image retrieval interfaces: A user perspective. In: *CIVR, Lecture Notes in Computer Science* (Vol. 3115, pp. 628–637). Springer.

Feng, S. L., Manmatha, R., & Lavrenko, V. (2004). Multiple Bernoulli relevance models for image and video annotation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2*, 1002–1009.

Goodrum, A. (2000). Image information retrieval: An overview of current research. *Information Science*, *3*, 2000.

Grigorescu, S. E., Petkov, N., & Kruizinga, P. (2002). Comparison of texture features based on Gabor filters. *IEEE Transactions on Image Processing, 11*(10), 1160–1167.

Grubinger, M., D, C. P., Henning, M., & Thomas, D. (2006). The iapr benchmark: A new evaluation resource for visual information systems. In: *International conference on language resources and evaluation, Genoa, Italy*.

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernatics*, *3*(6), 610–621. doi:10.1109/TSMC.1973.4309314.

Hironobu, Y. M., Takahashi, H., & Oka, R. (1999). Image-to-word transformation based on dividing and vector quantizing images with words. In: *Boltzmann machines, neural networks* (Vol. 4).

Huang, J., Kumar, S. R., Mitra, M., Zhu, W. J., & Zabih, R. (1997). Image indexing using color correlograms. In: *CVPR '97: Proceedings of the 1997 conference on computer vision and pattern recognition (CVPR '97)* (p. 762). IEEE Computer Society, Washington, DC, USA.

ICPR (2004). Ground truth database. University of Washington. http://www.cs.washington.edu/research/imagedatabase/groundtruth/. Accessed 17 May 2012.

Jin, Y., Khan, L., Wang, L., & Awad, M. (2005). Image annotations by combining multiple evidence & Wordnet. In: *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on multimedia*. ACM, New York, NY, USA.

Kwasnicka, H., & Paradowski, M. (2006). Multiple class machine learning approach for an image auto-annotation problem. In: *ISDA '06: Proceedings of the sixth international conference on intelligent systems design and applications* (pp. 347–352). IEEE Computer Society, Washington, DC, USA.

Kwasnicka, H., & Paradowski, M. (2008). Resulted word counts optimization-a new approach for better automatic image annotation. *Pattern Recognition, 41*(12), 3562–3571.

Lavrenko, V., Manmatha, R., & Jeon, J. (2003). A model for learning the semantics of pictures. In: *Advances in neural information processing systems NIPS 2003*. MIT Press.

Llorente, A., Motta, E., & Ruger, S. (2009). Image annotation refinement using Web-based keyword correlation. In: *SAMT '09: Proceedings of the 4th international conference on semantic and digital media technologies* (pp. 188–191). Springer-Verlag, Berlin, Heidelberg.

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

Lux, M., & Chatzichristofis, S. A. (2008). Lire: Lucene image retrieval: An extensible Java cbir library. In: *MM '08: Proceeding of the 16th ACM international conference on multimedia* (pp. 1085–1088). ACM, New York, NY, USA.

Makadia, A., Pavlovic, V., & Kumar, S. (2008). A new baseline for image annotation. In: *ECCV '08: Proceedings of the 10th European conference on computer vision* (pp. 316–329). Springer-Verlag, Berlin, Heidelberg.

Medvet, E., Bartoli, A., Davanzo, G., & De Lorenzo, A. (2011). Automatic face annotation in news images by mining the Web. In: *2011 IEEE/WIC/ACM international conference on Web intelligence and intelligent agent technology (WI-IAT)* (Vol. 1, pp. 47–54). doi:10.1109/WI-IAT.2011.101.

Michalak, K., Dzienkowski, B., Hudyma, E., & Stanek, M. (2011). Analysis of inter-rater agreement among human observers who judge image similarity. In: R. Burduk, M. Kurzynski, M. Wozniak, & A. Zolnierek (Eds.), *Computer recognition systems 4. Advances in intelligent and soft computing* (Vol. 95, pp. 249–258). Springer Berlin, Heidelberg.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., & Gool, L. (2005). A comparison of affine region detectors. *International Journal on Computer Vision*, *65*, 43–72. doi:10.1007/s11263-005-3848-x.

Nowak, S., & Huiskes, M. J. (2010). New strategies for image annotation: Overview of the photo annotation task at imageclef 2010. In: *CLEF (Notebook Papers/LABs/Workshops)*.

Nowak, S., Nagel, K., & Liebetrau, J. (2011). The clef 2011 photo annotation and concept-based retrieval tasks. In: *CLEF (Notebook Papers/Labs/Workshop)*.

Paradowski, M. (2008). Methods of image auto-annotation as an efficient tool for images describing (in Polish). Ph.D. thesis, Wrocław University of Technology.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 888–905.

Smeulders, A., Worring, M., Santini, S., Gupta, A., & Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machince Intelligence*, *22*(12), 1349–1380. doi:10.1109/34.895972.

Stanek, M., Broda, B., & Kwasnicka, H. (2010a). Patsi—Photo annotation through finding similar images with multivariate Gaussian models. In: *Computer vision and graphics—international conference ICCVG 2010(II)* (Vol. 6375, pp. 284–291). Springer, Lecture Notes in Computer Science.

Stanek, M., Broda, B., Paradowski, M., & Kwasnicka, H. (2010b). Magma—Efficient method for image annotation in low dimensional feature space based on multivariate Gaussian models. In: *Proceedings of the international multiconference on computer science and information technology* (pp. 131–138).

Tamura, H., Mori, S., & Yamawaki, T. (1978). Texture features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics, 6*, 460–473.

Tousch, A. M., Herbin, S., & Audibert, J. Y. (2012). Semantic hierarchies for image annotation: A survey. *Pattern Recognition, 45*, 333–345.

Verbeek, J., Guillaumin, M., Mensink, T., & Schmid, C. (2010). Image annotation with TagProp on the Mirflickr set. In: *Proceedings of the international conference on multimedia information retrieval*, *MIR '10* (pp 537–546). ACM, New York, NY, USA.

Wichert, A. (2008). Content-based image retrieval by hierarchical linear subspace method. *Journal of Intelligent Information Systems*, *31*, 85–107.

Zhang, D., Wong, A., Indrawan, M., & Lu, G. (2000). Content-based image retrieval using Gabor texture features. In: *IEEE transactions PAMI* (pp. 13–15).

Zhang, X., Li, Z., & Chao, W. (2011). Tagging image by merging multiple features in a integrated manner. *Journal of Intelligent Information Systems*. doi:10.1007/s10844-011-0184-1.