Semantic Spaces Revisited

Investigating the Performance of Auto-Annotation and Semantic Retrieval using Semantic Spaces

Jonathon S. Hare
jsh2@ecs.soton.ac.uk

Sina Samangooei
ss06r@ecs.soton.ac.uk

Paul H. Lewis
phl@ecs.soton.ac.uk

Mark S. Nixon
msn@ecs.soton.ac.uk

School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ
United Kingdom

ABSTRACT

Semantic spaces encode similarity relationships between objects as a function of position in a mathematical space. This paper discusses three different formulations for building semantic spaces which allow the automatic-annotation and semantic retrieval of images. The models discussed in this paper require that the image content be described in the form of a series of visual-terms, rather than as a continuous feature-vector. The paper also discusses how these term-based models compare to the latest state-of-the-art continuous feature models for auto-annotation and retrieval.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing;
H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance Evaluation*; I.4.9 [**Artificial Intelligence**]: Applications; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

Semantic Image Retrieval, Visual-terms, Evaluation, Semantic spaces, Latent Semantic Analysis, Probabilistic Latent Semantic Analysis

CIVR'08, July 7–9, 2008, Niagara Falls, Ontario, Canada.

1. INTRODUCTION

The generic notion of semantic image retrieval is one in which a corpus of images is made as accessible to retrieval and semantic understanding as a text corpus is now. This is an important notion that could affect everyone from professional multimedia searchers searching their archives, to home users searching their personal photo collections. The barrier to this notion is that whereas in text retrieval the methodology to index and retrieve documents is well understood in computational terms, an analogous methodology for multimedia is not. The biggest disappointment of over four decades worth of research into computational vision has been the general inability to transform images into sufficiently accessible information structures; this problem is known as the semantic gap in image retrieval.

Many different approaches to solving the problem of the semantic gap have been proposed; these almost invariably revolve around the idea of developing machines to automatically annotate images with keywords that can be used for indexing. Recently, it has been shown that whilst this technique is effective, it can often be better to record a keyword's similarity to an image, and use this similarity for ranked retrieval.

Perhaps the most obvious way to attempt to solve the problem of automatically annotating images is to attempt to use techniques from the computer vision field, in which the task of object recognition, or detection, is performed by specially trained classifiers which are used to determine the *class* or *keyword* of an object that has been segmented from the image. Unfortunately, this approach has limited use in the kind of real world images that we would like to deal with. The biggest problem is that the segmentation step is a chicken-and-egg problem; it is almost impossible to perform accurate object segmentation without any top-down knowledge of what the object looks like, but at the same time, the segmentation is required to describe what the object looks like.

Recent approaches to automatic annotation have tended to take a different approach, by inferring probabilities or similarities of concepts by analysing statistical properties of whole images. The basic premise of these automatic annotation approaches is that a model can be learned from a

training set of images that describes how low-level image features are related to higher-level keywords. This model can then be applied to unannotated images in order to automatically generate keywords that describe their content. In essence, the process of auto-annotation is analogous to translating from one language to another [1]. In fact, many of the state-of-the-art techniques for encoding low-level image content are based around the idea of transforming or quantizing the features to a vocabulary of visual-terms, which represent a purely visual language [13, 15, 30].

One of the earliest attempts at automatic annotation applied a co-occurrence model to keywords and low-level features of rectangular image regions [25]. Current techniques for auto-annotation generally fall into two categories; those that first segment images into regions, or 'blobs' and those that take a more scene-orientated approach, using global information. The segmentation approach has recently been pursued by a number of researchers. Duygulu *et al.*[8] proposed a method by which a machine translation model was applied to translate between keyword annotations and a discrete vocabulary of clustered 'blobs'. The data-set proposed by Duygulu *et al.*[8] has become a popular benchmark of annotation systems in the literature. Jeon *et al.*[18] improved on the results of Duygulu *et al.*[8] by recasting the problem as cross-lingual information retrieval and applying the Cross-Media Relevance Model (CMRM) to the annotation task. Jeon *et al.*[18] also showed that better (ranked) retrieval results could be obtained by using probabilistic annotation, rather than *hard* annotation. Lavrenko *et al.*[21] used the Continuous-space Relevance Model (CRM) to build continuous probability density functions to describe the process of generating blob features. The CRM model was shown to outperform the CMRM model significantly. Metzler and Manmatha [22] propose an inference network approach to link regions and their annotations; unseen images can be annotated by propagating belief through the network to the nodes representing keywords.

The models by Monay and Gatica-Perez [23], Feng *et al.*[10] and Jeon and Manmatha[19] use rectangular regions rather than blobs. Monay and Gatica-Perez [23] investigates Latent Space models of annotation using Latent Semantic Analysis and Probabilistic Latent Semantic Analysis, Feng *et al.*[10] use a multiple Bernoulli distribution to model the relationship between the blocks and keywords, whilst Jeon and Manmatha[19] use a machine translation approach based on Maximum Entropy. Blei and Jordan [3] describe an extension to Latent Dirichlet Allocation [4] which assumes that a mixture of latent factors are used to generate keywords and blob features. This approach is extended to multi-modal data in the article by Barnard *et al.*[1]. Most recently, Carneiro and Vasconcelos [5] have proposed a method that splits an image into blocks and models each keyword class as a hierarchical mixture of Gaussians describing the DCT coefficient information of these blocks. At the time of writing, Carneiro and Vasconcelos have the best published results on the dataset of Duygulu *et al* [8].

Oliva and Torralba [27, 28] explored a scene oriented approach to annotation in which they showed that basic scene annotations, such as 'buildings' and 'street' could be applied using relevant low-level global filters. Hare and Lewis [14] showed how vector-space representations of image content, created from local descriptors of salient regions within an image [12, 13, 30], could be used for auto-annotation by propagating semantics from similar images. Yavlinsky *et al.*[33] explored the possibility of using simple global features together with robust non-parametric density estimation using the technique of 'kernel smoothing'. The results shown by Yavlinsky *et al.*[33] were comparable with the inference network [22] and CRM [21]. Notably, Yavlinsky *et al.* showed that the Corel data-set proposed by Duygulu *et al.*[8] could be annotated remarkably well by just using global colour information. Hare and Lewis [15] also demonstrated that global histogram features could be used within a linear-algebraic retrieval/annotation system to yield better performance than Duygulu's machine translation approach combined with complex blob features for a number of different queries.

The system of Hare and Lewis [15] essentially constructs a special form of a vector space, called a semantic space, from the visual-terms and keyword annotations that occur in image documents. Unannotated images are projected into this space in order to be retrieved or annotated. The fundamental idea of a semantic space is that of a mathematical space in which objects are placed in a well defined pattern. The underlying principle is that objects that are *semantically similar* should be placed in such a way that they are near to each other in the space when measured using a suitable measure or metric. In terms of image retrieval research, such spaces are now becoming commonplace.

This paper revisits the linear-algebraic technique proposed in [15] and performs a number of evaluations using the Corel set with different image features. The linear-algebraic technique is also compared and contrasted to two probabilistic models based around Probabilistic Latent Semantic Analysis (PLSA).

2. LINEAR-ALGEBRAIC SEMANTIC SPACES

Our Linear-Algebraic Semantic Space approach [15, 11] is a generalisation of a text-retrieval technique called Cross Language Latent Semantic Indexing [20], which is itself an extension of Latent Semantic Indexing/Analysis (LSI/LSA) [6].

In general, any document (be it text, image, or even video) can be described by a series of observations, or measurements, made about its content. We refer to each of these observations as terms. Terms describing a document can be arranged in a vector of term occurrences, i.e. a vector whose i-th element contains a count of the number of times the i-th term occurs in the document. There is nothing stopping a term vector having terms from a number of different modalities. For example a term vector could contain term-occurrence information for both 'visual' terms and textual annotation terms. Given a corpus of documents, it is possible to form a matrix of observations or measurements (i.e. a term-document matrix).

Fundamentally, the Semantic Space technique works by estimating a rank-reduced factorisation of a term-document matrix of data, $\mathbf{O}$, into a term matrix $\mathbf{T}$ and a document matrix $\mathbf{D}$:

$$\mathbf{O} \approx \mathbf{TD} \ . \tag{1}$$

The two vector bases created in the decomposition form aligned vector-spaces of terms and documents. The rows of the term matrix, $\mathbf{T}$, create a basis representing a position in the space of each of the observed terms. The columns of

the document matrix, $\mathbf{D}$, represent positions of the observed documents in the space. Similar documents and terms share similar locations in the space.

The Singular Value Decomposition proves to be a useful tool for estimating the factorisation in Equation 1. The term-document matrix, $\mathbf{O}$ can be decomposed using SVD into a $m \times r$ matrix $\mathbf{U}$, a $r \times r$ diagonal matrix $\mathbf{\Sigma}$ and a $r \times n$ matrix $\mathbf{V}^T$,

$$\mathbf{O} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \ , \tag{2}$$

such that $\mathbf{U}^T\mathbf{U} = \mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathcal{I}$, where $\mathcal{I}$ is the identity matrix. Now partitioning the $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{V}^T$ matrices as follows:

$$\begin{aligned}
\mathbf{U} &= \underbrace{\left[\ \mathbf{U}_k\ \right.}_{k}\underbrace{\left|\ \mathbf{U}_N\ \right]}_{r-k}\ \}m \\
\mathbf{\Sigma} &= \left[\begin{array}{c|c} \mathbf{\Sigma}_k & 0 \\ \hline 0 & \mathbf{\Sigma}_N \end{array}\right] \begin{array}{l} \}k \\ \}r-k \end{array} \\
&\quad\ \underbrace{\phantom{\mathbf{\Sigma}_k}}_{k}\ \underbrace{\phantom{\mathbf{\Sigma}_N}}_{r-k} \\
\mathbf{V}^T &= \underbrace{\left[\begin{array}{c} \mathbf{V}_k^T \\ \hline \mathbf{V}_N^T \end{array}\right.}_{n} \begin{array}{l} \}k \\ \}r-k \end{array} \ ,
\end{aligned} \tag{3}$$

we have, $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T + \mathbf{U}_N\mathbf{\Sigma}_N\mathbf{V}_N^T$. If we now assume that there are k independent terms in the term-document matrix $\mathbf{O}$, then it can be shown that the best possible rank-k approximation (in the least-squares sense) to $\mathbf{O}$ is given by $\mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T$.

Since the matrix $\mathbf{O}$ contains all of the relationships between terms and documents, it follows that all co-occurrences between terms can be characterised by $\mathbf{O}\mathbf{O}^T$ and between documents by $\mathbf{O}^T\mathbf{O}$. Expanding using the SVD expression (2), it follows that:

$$\begin{array}{rclcl} \mathbf{O}\mathbf{O}^T &=& (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^T &=& \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T \ , \\ \mathbf{O}^T\mathbf{O} &=& (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^T(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T) &=& \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T \ . \end{array} \tag{4}$$

From this, it can be seen that the co-occurrence relationships between terms is independent of $\mathbf{V}$ and conversely, the document co-occurrence is independent of $\mathbf{U}$. This implies that individual terms are represented by rows of $\mathbf{U}$, and documents by columns of $\mathbf{V}^T$.

Assume that we have two collections of images; a training set with keyword annotations and a test set without. The content of each image can be represented by a vector of 'visual-term' occurrences. A cross-modality term-document matrix, $\mathbf{O}_{train}$ can be created for the training set of images by combining the visual-term occurrence vector with the keyword-term occurrence vector for each image. This can then be factorised according to Equation 1 into a term matrix $\mathbf{T}_{train}$ and a document matrix $\mathbf{D}_{train}$ by using the singular value decomposition and letting $\mathbf{T} = \mathbf{U}_k$ and $\mathbf{D} = \mathbf{\Sigma}_k\mathbf{V}_k^T$. As described above, the rows of $\mathbf{T}$ describe the position in the space of each term, and the columns of $\mathbf{D}$ describe the position of each document.

In order to make the unannotated test images searchable, we can project them into the semantic space described by $\mathbf{T}_{train}$ (and $\mathbf{D}_{train}$). Firstly, a cross-modality term-document matrix, $\mathbf{O}_{test}$ must be created for the test set of images by setting the number of occurrences of each (unknown) keyword to 0. It can be shown that it is possible to create a document matrix, $\mathbf{D}_{test}$ for the test documents as follows:

$$\mathbf{D}_{test} = \mathbf{T}_{train}^{\mathrm{T}}\mathbf{O}_{test} \ . \tag{5}$$

In order to query the test set for images relevant to a term, we just need to rank all of the images based on their position in the space with respect to the position of the query term in the space. The angle between the vectors or cosine similarity is a suitable measure for this task.

2.1 Weighting the term-document matrix

In text-retrieval it is often beneficial to apply some form of weighting to individual terms and whole documents in order to provide some normalisation. The same may also be true of using this technique, and is perhaps even more important given the possible issues related to the difference in magnitudes of the counts of the visual-terms compared to the semantic terms. It has been suggested that a suitable method for normalising the data is to use a normalised-entropy scheme [2], which normalises the document length and word entropy. The normalised-entropy weighting can be expressed as follows: for each element (i, j) of $\mathbf{O}$,

$$w_{i,j} = (1 - \varepsilon_i)\frac{c_{i,j}}{n_j} \ , \tag{6}$$

where,

$c_{i,j}$	number of times w_i occurs in d_j
n_j	total number of words in d_j
ε_i	normalised entropy of w_i in the set of documents

The global weighting $(1 - \varepsilon_i)$ reflects the fact that two words in the same document, with the same count, do not necessarily convey the same amount of information about the document, and thus is a measure of the indexing power of a given term. The intuition is that a word that is distributed across many documents will have a lower indexing power than a word that only occurs in a few specific documents. If we denote the number of times the total number of times a word w_i occurs in all documents by $t_i = \sum_j c_{i,j}$, the expression for ε_i is

$$\varepsilon_i = \frac{1}{\log N}\sum_{j=1}^{N}\frac{c_{i,j}}{t_i}\log\frac{c_{i,j}}{n_j} \ . \tag{7}$$

2.2 Experiments

Our previous work [15, 16] demonstrated that the linear-algebraic technique can be quite effective at image retrieval. One of the problems with many of the recent works on automatic annotation is that whilst they have used the Corel dataset as a benchmark from which to compare results, each technique has used a different type of visual feature. The use of different features makes it rather difficult to compare algorithms directly, as it it impossible to say what effect the choice of feature has on the machine learning performed by the algorithm; as an example, it is difficult to say whether the impressive results of [5] are due to the performance of the probabilistic model, or to the power of the DCT features used [32].

In this paper we investigate the performance of the technique for semantic retrieval and auto-annotation tasks using the Corel dataset [8] using a number of different types of image features. Results from these experiments allow us to

quantify the performance of our technique when compared to other techniques, but also to say something about the expressive power of the different visual features.

The Corel dataset has been criticised in the past as both being "too easy", and as too small for proper retrieval evaluation [31, 26]. However, that being said, it is still used as the *defacto* standard in most auto-annotation papers. In this study, we believe that the choice of dataset is reasonable because the experiments will be repeatable and comparable. Also, we don't believe that the dataset is quite as easy as has sometimes been suggested since the state-of-the-art techniques struggle to annotate effectively. One answer to this is that the dataset is actually quite representative of other real-world datasets in that it contains many errors, and strange keyword choices. These factors confound the problem of training a machine to learn how to annotate image content effectively, but are realistic of training data in the real world.

We have split the dataset into three subsets for experimental purposes; a 4000 image training set, a 500 image validation set, and a 500 image test set. The 500 image test set is the same as used in [8]. The linear-algebraic technique described above has one parameter in which to optimise; the number of dimensions, or k. This is optimised by training on the training set, and finding the value of k that maximises the mean average precision of a hypothetical semantic retrieval scenario using the validation set [15]. Once the optimal k value has been found, a new space is trained using the training and validation sets combined before the unannotated test set is projected in.

2.2.1 Image features

In order to investigate the effect of different feature morphologies on the performance of the linear-algebraic technique, we have selected four different types of visual-term feature. In particular, we study a feature based on the discrete cosine transform in detail.

Global RGB Histogram.

The first type of feature selected is a 64-bin (4×4×4) global RGB histogram. This is the same feature that was demonstrated to perform surprisingly well on the Corel dataset in [15]. Each bin of the histogram is considered to represent a single visual term, and the size of the bin represents the number of times that visual term occurs in the image.

Local RGB Histogram.

The global colour histogram described above completely discards all information about the layout of colour in the image. If we first segment the image into blocks, and then calculate a colour histogram for each block, it is possible to develop a rudimentary descriptor that describes the colour at a rough location in the image. For our experiments, we split each image into 16 blocks (four evenly sized intervals along each axis) and calculated a 64-bin RGB histogram for each block. Each histogram bin at each of the 16 different locations was taken to be a different visual term, and as before the size of the bin represented the number of occurrences of the respective term.

Blobs.

The blobs feature is the same as found in [8]. The feature was created by segmenting each image into a number of blobs, and then calculating a descriptor using various colour, texture and shape attributes of each blob in the respective image. The set of descriptors was then clustered using k-means to create a vocabulary of 500 visual-terms. Each blob was then converted to a visual term by vector quantising its descriptor into the nearest visual term. Term-occurrence vectors were finally calculated by counting the occurrences of each visual term in each image.

Figure 1: Example of a visual vocabulary from clustered DCT blocks

Clustered DCT Features.

It has been suggested that the reason for the impressive performance of the system presented in [5] is partially due to the Discrete Cosine Transform (DCT) features used. We can use quite the same feature as [5] in our annotation/retrieval system because of the requirement that we have discrete visual-terms, rather than a continuous feature; however, it is possible to use the DCT to create a visual vocabulary by clustering image blocks in the DCT domain, and then applying vector quantisation to create lists of visual-terms for each image.

In our implementation, we split each image into a sequence of 8×8 blocks. We also left a 4-pixel border around the edge of each image in order to reduce the likelihood of problems occurring due to the black borders in many of the Corel images. For each of the Red, Green and Blue planes of the block we calculated the DCT, and ordered the DCT coefficients from highest to lowest frequency. It is well known that the lowest frequency coefficients are less important visually, so of the 64 DCT coefficients, we kept only the highest 10 coefficients from each plane (including the DC coefficient). The selected coefficients from each plane were appended together to form a feature vector for the respective image block.

Once sets of feature-vectors had been calculated for each image, a random sample was drawn and clustered using K-means. The cluster centres formed a codebook, or vocabulary, of visual-terms which was then used to assign a visual term to each feature-vector by finding the closest term in the codebook (using Euclidean distance). An example of the representative image blocks found in a typical 500 term vocabulary generated from the Corel set is shown in Figure 1.

There are a number of factors that affect the quality of image description using this technique; for example, how many clusters (and thus visual-terms) should be defined. This issue, and a number of others are investigated below.

Number of clustered features.

The number of clusters chosen is an important factor. In order to determine an optimal vocabulary size we created a number of different codebooks of varying sizes (500, 1000, 3000 and 6000 terms). For each of these vocabularies, we

trained spaces on the training data and used the validation data as a basis for performing retrieval experiments. The mean average precision over a range of different k values is plotted in Figure 2.

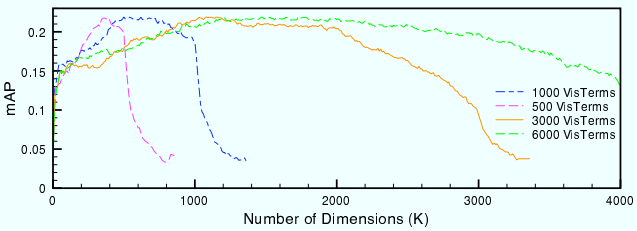

Figure 2: Plot showing variation in mean average precision (averaged over all queries) versus the number of dimensions for 4 different sizes of visual-term codebook.

Interestingly, Figure 2 shows that the size of the codebook has negligible effect on the maximally achievable mean average precision. The only noticeable effect is that the value of k for which mAP is maximised increases with vocabulary size. This is not really surprising as we expect that the larger vocabulary sizes will introduce more independent terms. For practical purposes, we choose to use the smallest, 500 term, vocabulary in subsequent experiments because it is more computationally efficient to deal with a smaller vocabulary, and there is no trade-off in retrieval performance for doing so.

Effect of clustering on DCT features.

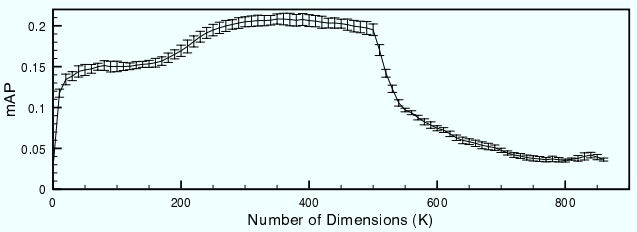

Figure 3: Plot showing variation in mean average precision (averaged over all queries) versus the number of dimensions. The error bars show 1 standard deviation of mAP over 11 different 500 visual-term codebooks.

One possible problem with our approach to clustering the DCT feature-vectors is that the clustering has a certain amount of randomness and uncertainty associated with it; firstly we are randomly selecting a subset of points from which to cluster, and secondly, k-means starts at random points. It is interesting to investigate how much of a problem this causes. Figure 3 shows the average and standard deviation of the mAP over a range of k for a number 500 term codebooks. The standard deviation is fairly consistent across the whole range of k. On average the standard deviation of mAP from the different codebooks is little over 0.6%, which shows that the randomness of the clustering is almost negligible; that being said, we choose to use the best performing codebook for the remainder of the experiments. It should however be noted that there is no guarantee that the codebook which provides the best performance on the validation data will work best on the actual test data.

Overlapping DCT blocks.

In the work of Carneiro and Vasconcelos [5], overlapping DCT blocks are selected using a sliding window scheme. Overlapping blocks will increase the number of visual-terms extracted from a given image. In order to investigate whether the overlapping of blocks helps performance, we created 500-term vocabularies and semantic spaces for varying amounts of offset between subsequent image blocks. Offsets of 2, 4, 6, and 8 pixels were used. The 8 pixel offset corresponds to no overlap.

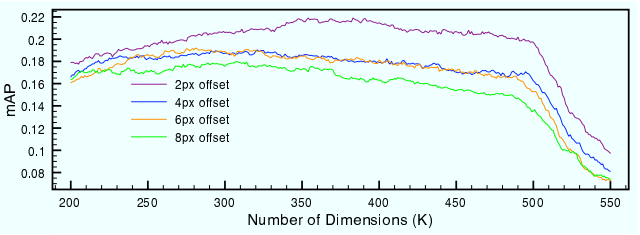

Figure 4: Plot showing variation in mean average precision (averaged over all queries) versus the number of dimensions for 4 different amounts of offset of the extracted 8×8 image blocks.

Figure 4 shows the effect the varying offset has on the mAP over a range of k values. Interestingly, an offset of 2-pixels (the most overlap of blocks) leads to a much greater maximum mAP than with lower amounts of overlap. As mentioned earlier, increased amounts of overlap result in more visual-terms for a given image; it is likely that this in turn leads to a much richer image description, which then leads to an improved retrieval performance.

2.2.2 Retrieval Performance

Semantic retrieval performance of the different features is compared by creating semantic spaces for each feature and using the trained keywords to attempt retrieval of the images in the test set. Since the ground truth annotations of the test set are known, it is possible to determine which images are relevant to a particular keyword query and thus calculate precision and recall.

Figure 5 illustrates the performance of each of the features, with and without weighting, within the semantic space framework. Table 1 summarises the mean average precision for the retrieval experiments. It should be noted that these precision-recall scores are perhaps a little misleading as the system does not perform equally for all queries; that is to say that for each of the different features, certain queries will perform much better than others, depending on what underlying relationships have been learnt by the system.

The results show a number of findings; firstly, the clustered DCT feature does indeed perform better than the other features. Secondly, and rather surprisingly, the local RGB Histogram feature outperforms the Blob feature (significantly, if we just consider the unweighted variants). Thirdly, the use of the weighting scheme helps in some cases, but hinders in others. More discussion of these findings can be found in Section 4.

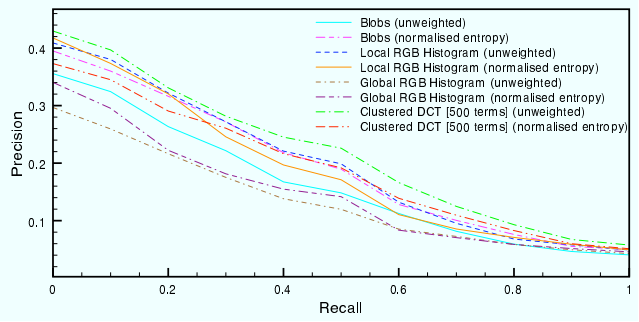

Figure 5: Interpolated precision-recall curves for semantic retrieval using each of the different image features, with and without weighting

Table 1: Summary of Retrieval Performance using linear-algebraic method

Feature	unweighted		normEnt	
	K_{opt}	mAP	K_{opt}	mAP
Global RGB Histogram	39	0.126	27	0.137
Local RGB Histogram	164	0.187	223	0.178
Blobs	39	0.155	21	0.184
Clustered DCT Feature	354	0.208	345	0.181

2.2.3 *Annotation Performance*

In addition to performing semantic search by locating a keyword in the space and then finding images sharing a similar location, the semantic space technique can be used in reverse; an un-annotated image can be located in the space, and similar keywords can be found, thus allowing automatic annotation to take place.

Following the methodology of [5], for each image in the test set we predict the annotations of each image to be the closest 5 keywords in the semantic space. Performance of the auto-annotation can be calculated as follows; For each keyword, assume there are w_H images with that keyword as ground truth, and that the system has annotated w_{auto}, of which w_C are correct. It is then possible to determine a precision and recall measure for each keyword by $precision = \frac{w_C}{w_{auto}}$ and $recall = \frac{w_C}{w_H}$. Also, considering the number of keywords with a nonzero recall (i.e. $w_C > 0$) gives an indication of how many keywords have been effectively learnt.

Table 2 shows the annotation precision and recall averaged over all query terms, together with the number of learnt terms. Again, the clustered DCT performs well. However the most surprising result is that the local RGB histogram is able to learn 100 terms (albeit with lower precision and recall than the DCT feature). The effect of the weighting is also interesting — again, it improves some features slightly, but hinders others.

3. PROBABILISTIC SEMANTIC SPACES

To highlight the performance of our linear-algebraic LSA-based approach we compare its results with those of a related, yet inherently different, semantic correlation approach. Probabilistic Latent Semantic Analysis (PLSA)[17] is a technique originally designed to solve the problem of automatic classification of text documents through word-document co-occurrence. However, in contrast to LSA, which stems from linear algebra, PLSA uses a statistical methods to formulate and solve the problem.

The key concept in PLSA is that of several generative latent *classes*, to which the presence of particular terms in a document can be attributed. That is to say, the assumption of the existence of N_z classes $z \in \{z_1, ..., z_{N_z}\}$ which are *underlying concepts* that generate terms $t \in \{t_1, ..., t_{N_t}\}$ and documents $d \in \{d_1, ..., d_{N_d}\}$. These generative classes can be interpreted as the corpus' *subjects* and are the tool used to associate terms and documents which have a similar *meaning* but possibly no direct co-occurence. This is achieved by maximising the probability of the existence of observed terms and documents given the underlying concept.

Let $P(d_i)$ be the probability of choosing a particular document. $P(z_k|d_i)$ is therefore the probability of a latent class given a particular document and $P(t_j|z_k)$ is the probability of a particular term occurring given a class. Using the product rule, the probability of an observable pair $P(d_i, t_j)$ can be expressed as:

$$P(d_i, t_j) = P(d_i)P(t_j|d_i) \ , \tag{8}$$

where,

$$P(t_j|d_i) = \sum_{k=1}^{K} P(t_j|z_k)P(z_k|d_i) \ . \tag{9}$$

This makes the assumption that term t and document d are generated conditionally independently given a particular class z. Subsequently, the goal becomes the estimation of $P(d_i)$,$P(t_j|z_k)$ and $P(z_k|d_i)$ such that we maximise the probability of document and term pairs we have already observed, i.e. the maximisation of $\mathcal{L}$ given that:

$$\mathcal{L} = \sum_{i=1}^{N_t}\sum_{j=1}^{N_d} n(d_j, t_i) log P(d_j, t_i) \ , \tag{10}$$

where $n(d_j, t_i)$ is the number of times a document and term pair have been observed or simply put, the occurrences term t_j in a document d_i.

An approach for maximising $\mathcal{L}$ is the iterative Expectation Maximisation (EM)[7] algorithm. EM works by applying 2 steps alternately: (i) an expectation step (E) where the posterior probability of a class given all observed document and feature pairs $P(z_k|d_i, t_j)$ is recalculated (ii) and maximisation (M) where the posterior probabilities used as parameters in step (i) are updated using the new posterior probabilities. From repeated application of Bayes rule to $P(z|d, t)$ we can arrive at the (E) step which is:

$$P(z_k|d_i, t_j) = \frac{P(t_j|z_k)P(z_k|d_i)}{\sum_{k=1}^{N_z} P(t_j|z_k)P(z_k|d_i)} \ . \tag{11}$$

Also, using marginalisation and conditioning on the $P(t|z)$ and $P(z|d)$ we can arrive at the (M) step which is:

$$P(t_j|z_k) = \frac{\sum_{i=1}^{N_d} n(t_j|d_i)P(z_k|d_i, t_j)}{\sum_{m=1}^{N_t}\sum_{i=1}^{N_d} n(t_m, d_i)P(z_k|d_i, t_m)} \tag{12}$$

$$P(z_k|d_i) = \frac{\sum_{i=1}^{N_t} n(t_j, d_i)P(z_k|d_i, t_j)}{n(d_i)} \ . \tag{13}$$

Algorithm 1 illustrates the EM process. The algorithmic form shows more clearly that $P(t_j|z_k)$ and $P(z_k|d_i)$ are in

Table 2: Summary of Annotation Performance using linear-algebraic method

Feature	unweighted			normEnt		
	Precision	*Recall*	*#words(rec>0)*	*Precision*	*Recall*	*#words(rec>0)*
Global RGB Histogram	0.109	0.101	55	0.144	0.149	86
Local RGB Histogram	0.187	0.211	100	0.154	0.195	88
Blobs	0.125	0.118	65	0.161	0.154	87
Clustered DCT Feature	0.192	0.246	97	0.191	0.239	98

Algorithm 1 EM Algorithm Pseudo-code

```
{Start with initial random probabilities}
P(t|z) = rand(N_t, N_z)
P(z|d) = rand(N_z, N_d)
for I iterations do
    {The E step, find P(z|d,t) for the current values}
    for Each class z_k do
        for Each each observation (d_i, t_j) do
            P(z_k|d_i, f_j) Calculated from current P(t_j|z_k) and
            P(z_k|d_i)
        end for
    end for
    {The M step, using new P(z|d,t) find new values}
    {Find the new P(t|z) and P(z|d)}
    for Each class z_k do
        for Each word d_i do
            for Each class t_j do
                P(t_j|z_k) Calculated using new P(z_k|d_i, t_j)
                P(z_k|d_i) Calculated using new P(z_k|d_i, t_j)
            end for
        end for
    end for
end for
```

fact 2 matrices of size (N_f, N_z) and (N_z, N_d) respectively. It is also interesting that the probability $P(d_i, t_j)$ for any give d_i, w_j is in fact the dot product of the j^{th} row in $P(t_j|z_k)$ with the i^{th} column in $P(z_k|d_i)$ scaled by $P(d_i)$, making the practical process of image retrieval and comparison using PLSA quite similar to LSA using SVD.

The outputs of the PLSA iterative EM process can be directly applied to the image retrieval tasks of (i) document by term (ii) annotation of an unseen document.

Both these retrieval tasks can be approached in a similar way to LSA's projection technique. However, rather than projecting a query document into the term space, or finding the geometrically close terms to annotate a document, the unseen document's $P(z_k|d_{unseen})$ distribution is discovered using further iterations of EM combined with a process called *folding*, such that d_{unseen} is the unobserved document or term query constructed as a complete or partial document. Having run EM, we know the distribution of $P(t_j|z_k)$; folding simply involves keeping this distribution fixed after each M step and discovering a new $P(z_k|d_i)$ based on the observed instances in the query document. This process guarantees the $P(z_k|d_i)$ distribution of d_{unseen} which most correctly matches the previously learnt class structure. In the case of retrieval, the query $P(z_k|d_i)$ can be directly compared to existing $P(z_k|d_i)$ and comparisons made to existing documents using some distance metric (e.g. cosine or Euclidian). In the case of annotation, $P(t_j|d_{unseen})$ can be utilised as shown:

$$P(t_j|d_{unseen}) = \sum_{k=1}^{N_z} P(t_j|z_k)P(z_k|d_{unseen}) \ . \quad (14)$$

Which can be calculated using the dot product of each row in $P(t_j|z_k)$ with the $P(z_k|d_{unseen})$, the highest values are the most probable annotations.

PLSA-Mixed vs PLSA-Word

It should be noted that the original development of PLSA didn't expect the terms of the domain to be of more than one type, where as, in the case of image retrieval or annotation, there are visual-terms combined with separate semantic terms. It has been noted [24] that there is no clear reason to believe these 2 distinct types of term are equally important in defining the underlying concepts and so should be used to define the concept space simultaneously. Rather, approaches have been taken to improve recognition results by acknowledging the greater importance of words to define the underlying concepts.

The basic approach for applying PLSA to visual and semantic terms is to simply concatenate the terms into the same bag-of-words representation. This scheme has been dubbed PLSA-Mixed and produces comparatively poor results. PLSA-Words is a scheme with better results that acknowledges that the semantic terms are more likely to define the structure of the underlying *subjects* which the classes z_k represent. Subsequently the semantic variables are used to initially learn the $P(z_k|d_i)$ distribution. Once this is found, a similar technique to folding is utilised and the $P(z_k|d_i)$ is kept fixed while the $P(t_j|z_k)$ distribution is discovered utilising the visual-terms. This has a dual affect of improving recognition and annotation results, as well as increasing the speed of the algorithm as the iterative scheme takes less time to settle once the initial class structure is known, which itself is found very quickly as there are fewer semantic terms than there are visual.

Results for both PLSA-Mixed and PLSA-Words are shown in the following section

3.1 Experiments

Performance experiments for PLSA used 4000 training images, 500 verification and 500 test images all from the Corel set. The training set was fully annotated, the validation set used for finding correct parameters and the test set used for final results on a new training set which combines the validation and training sets. The visual-terms utilised were the optimal K-Means clustered DCT features described in Section 2.2.1.

Firstly, an experiment was performed using the validation set to gauge the number of underlying classes, as this is unknown for this document corpus. This was performed using both the PLSA-Mixed and PLSA-Word approaches.

For each class size between 10 and 800, 10 attempts are made to find mAP to account for the random nature of the iterative EM algorithm. Once N_z was ascertained, we used the best number of classes to gauge the likelihood of each semantic term for each document in the test set. Also the top 5 semantic terms were ascertained to allow for comparison to SVD using the technique mentioned in Section 2.2.3. This was done against a training set comprised of 4500 documents in the training and validation sets combined. The results can be seen below:

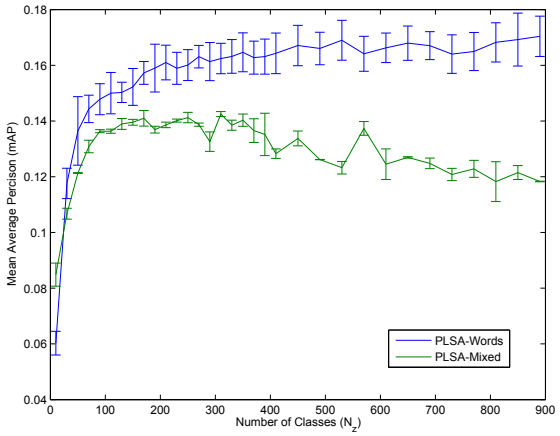

Figure 6: A Graph showing mean Average Precision of PLSA-Words and PLSA-Mixed

The results in Figure 6 show that after roughly $N_z = 150$ classes are added, there is a levelling out of precision. The point at which the two lines level out is likely to be the true N_z, i.e. the correct number of *subjects* that exist in the corpus. Also, due to the independence of individual classes, the addition of classes doesn't reduce precision, but rather the iterative EM algorithm guarantees classes randomly initiated to roughly the same values will be grouped together, i.e. assigned similar $P(z_k|t_j, d_i)$ values. We ascribe the slight dipping of the PLSA-Mixed results to insufficient iterations of the EM algorithm.

Figure 7 shows the precision-recall curves for the semantic retrieval experiment (Section 2.2.2) using the PLSA-based semantic space techniques. Table 3 shows the retrieval and auto-annotation scores for PLSA-Mixed and PLSA-Words, together with the linear-algebraic model when using optimal DCT terms.

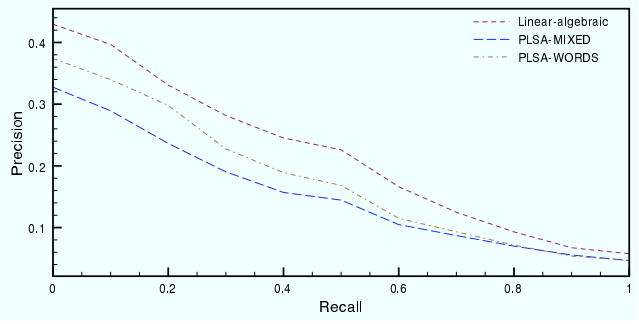

Figure 7: Precision-Recall curves for retrieval experiments using the PLSA-Words, PLSA-Mixed, and Linear-algebraic techniques.

4. DISCUSSION

Section 2 and Section 3 have shown three approaches to semantic annotation. There are however several points of discussion in the results shown and experiments performed.

Comparison to other annotation techniques

It is interesting to compare the performance of the automatic annotations methods presented in this paper with other recent and state-of-the automatic annotation techniques. Table 4 summarises the annotation performance of a number of techniques. In particular, Table 4 shows results from Mori et al.'s Co-occurrence model [25], Duygulu *el at.*'s machine translation approach [8], the Cross Media Relevance Model (CMRM) [18], the Continuous-space Relevance Model (CRM) [21], the Multiple Bernoulli Relevance Model (MBRM) [10] and Supervised Multiclass Labelling (SML) [5], in addition to our results from using PLSA-Words and Linear Algebraic Semantic Spaces (LASS). The results shown in Table 4 shows that the techniques presented in this paper fall in the mid-range of the other techniques. One interesting facet is that our techniques are the best performing of any techniques that use discrete visual-term features; the techniques that outperform ours use continuous feature-spaces.

Weighting

The results for LSA show that with certain visual-terms weighting greatly improves results while with others it damages results. We hypothesise that this is due to differences in the amount of information different visual features contain. In one situation the relatively minimal occurrence of a certain visual feature may imply lots of information, whereas for another visual feature this may have no relevance at all, yet weighting assumes identical information based only on appearance. For example, take the colour "Purple" in the colour histogram and the blob containing the concept "Tiger". Both may appear relatively little throughout the corpus, "purple" only once on a jockey's uniform and "tiger" in a few pictures of tigers. However, the appearance of purple doesn't necessarily imply jockey where the existence of "tiger blob" is a pretty good indicator for tiger.

PLSA versus LSA

The results in Table 3 confirm several notions. Firstly, PLSA-Words achieves better results than PLSA-Mixed showing that indeed corpus concepts are more appropriately derived from semantic terms than visual-terms. However, contrary to [24] who reported superior performance of PLSA-Words, our results show worse results when compared to LSA in the form of our linear-algebraic technique. We ascribe this difference partially to the different features utilised in their experiments, namely those which take structural features such as those obtained from SIFT into account. Also, their LSA results utilise an approach called SVD-Cos[29], developed primarily for blob annotation rather than whole image annotation; giving further explanation for their poor LSA based global annotation results.

Table 3: Final Comparative Test Set Results

	Retrieval	Auto-Annotation		
Scheme	*mAP*	*precision*	*recall*	*#words(rec>0)*
PLSA-Mixed	14.1%	7.7%	11.2%	49
PLSA-Words	16.2%	13.7%	17.9%	80
Linear-algebraic	20.1%	19.2%	21.1%	97

Table 4: Annotation performance of various other automatic annotation techniques compare to the linear-algebraic technique and PLSA-Words

Models	*Co-Occurrence*	*Translation*	*CMRM*	**PLSA-Words**	**LASS**	*CRM*	*MBRM*	*SML*
Words with Recall > 0	19	49	66	80	97	107	122	137
Mean Per-word Recall	0.02	0.04	0.09	0.18	0.21	0.19	0.25	0.29
Mean Per-word Precision	0.03	0.06	0.10	0.14	0.19	0.26	0.24	0.23

5. CONCLUSIONS AND FUTURE WORK

We have shown a comparative set of results of three methods for building semantic spaces for the Corel image dataset. We have also shown the usefulness of various discrete visual term approaches and compared their results, both with and without an entropy based weighting scheme. Our results show that an approach using a DCT to compute a set of K-Means optimal blocks gave the best results and that our linear-algebraic technique gave the best precision and recall over both PLSA-Mixed and PLSA-Words, regardless of their more rigourous theoretical grounding. Both the linear-algebraic and PLSA-Words techniques outperform any currently known automatic-annotation methodology that uses discrete visual-terms as a basis for image description. The following describes some of the future directions in which we intend to investigate.

Optimising K on a per-term basis

Some preliminary experiments which involve finding the optimal SVD rank for each semantic-term showed that each term optimised at different values of K for the chosen DCT configuration. One approach would be to find the K rank value for each term that most improves its annotation precision. This avenue of research would be effectively applying feature subset selection techniques to the set of singular values finding the set of projected document and term features which best describe the space for a given scenario.

Spatially consistent models

The impressive performance of the local RGB histogram indicates that the spacial information provided by a visual term might be important. Subsequently the addition of spatial information to the more sophisticated visual-terms may result in improved annotation.

Semantic term significance

In the Corel dataset there is no indication of the relevance of a particular annotation, rather all annotations are considered equal. Often this is not the case and there may be one or two annotations which are the main subject of the image and others which are less so. Such information could be correlated with the existence of visual-terms and improve the information they provide

6. ACKNOWLEDGMENTS

The third author is grateful for the support of EU funding under the OpenKnowledge and HealthAgents grants numbered IST-FP6-027253 and IST-FP6-027213.

7. REFERENCES

[1] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *J. Mach. Learn. Res.*, 3:1107–1135, 2003.

[2] J. R. Bellegarda. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, 88(8):1279–1296, Aug. 2000.

[3] D. M. Blei and M. I. Jordan. Modeling annotated data. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134, New York, NY, USA, 2003. ACM Press.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

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

[6] 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.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[8] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 97–112, London, UK, 2002. Springer-Verlag.

[9] P. G. B. Enser, Y. Kompatsiaris, N. E. O'Connor, A. F. Smeaton, and A. W. M. Smeulders, editors. *Image and Video Retrieval: Third International Conference, CIVR 2004, Dublin, Ireland, July 21-23, 2004. Proceedings*, volume 3115 of *Lecture Notes in Computer Science*. Springer, 2004.

[10] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *CVPR (2)*, pages 1002–1009, 2004.

[11] J. S. Hare. *Saliency for Image Description and Retrieval*. PhD thesis, University of Southampton, 2005.

[12] J. S. Hare and P. H. Lewis. Salient regions for query by image content. In Enser et al. [9], pages 317–325.

[13] J. S. Hare and P. H. Lewis. On image retrieval using salient regions with vector-spaces and latent semantics. In W. K. Leow, M. S. Lew, T.-S. Chua, W.-Y. Ma, L. Chaisorn, and E. M. Bakker, editors, *CIVR*, volume 3568 of *Lecture Notes in Computer Science*, pages 540–549. Springer, 2005.

[14] J. S. Hare and P. H. Lewis. Saliency-based models of image content and their application to auto-annotation by semantic propagation. In *Proceedings of the Second European Semantic Web Conference (ESWC2005)*, Heraklion, Crete, May 2005.

[15] J. S. Hare, P. H. Lewis, P. G. B. Enser, and C. J. Sandom. A Linear-Algebraic Technique with an Application in Semantic Image Retrieval. In H. Sundaram, M. Naphade, J. R. Smith, and Y. Rui, editors, *Image and Video Retrieval, 5th International Conference, CIVR 2006, Tempe, AZ, USA, July 2006, Proceedings*, volume 4071 of *Lecture Notes in Computer Science*, pages 31–40. Springer, 2006.

[16] J. S. Hare, P. H. Lewis, P. G. B. Enser, and C. J. Sandom. Semantic facets: an in-depth analysis of a semantic image retrieval system. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 250–257, New York, NY, USA, 2007. ACM Press.

[17] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.*, 42(1-2):177–196, 2001.

[18] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 119–126, New York, NY, USA, 2003. ACM Press.

[19] J. Jeon and R. Manmatha. Using maximum entropy for automatic image annotation. In Enser et al. [9], pages 24–32.

[20] T. K. Landauer and M. L. Littman. Fully automatic cross-language document retrieval using latent semantic indexing. In *Proceedings of the Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research*, pages 31–38, UW Centre for the New OED and Text Research, Waterloo, Ontario, Canada, October 1990.

[21] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[22] D. Metzler and R. Manmatha. An inference network approach to image retrieval. In Enser et al. [9], pages 42–50.

[23] F. Monay and D. Gatica-Perez. On image auto-annotation with latent space models. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 275–278. ACM Press, 2003.

[24] F. Monay and D. Gatica-Perez. Modeling semantic aspects for cross-media image indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1802–1817, 2007.

[25] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *Proceedings of the First International Workshop on Multimedia Intelligent Storage and Retrieval Management (MISRM'99)*, 1999.

[26] H. Müller, S. Marchand-Maillet, and T. Pun. The truth about corel - evaluation in image retrieval. In M. S. Lew, N. Sebe, and J. P. Eakins, editors, *CIVR*, volume 2383 of *Lecture Notes in Computer Science*, pages 38–49. Springer, 2002.

[27] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, 2001.

[28] A. Oliva and A. B. Torralba. Scene-centered description from spatial envelope properties. In *BMCV '02: Proceedings of the Second International Workshop on Biologically Motivated Computer Vision*, pages 263–272, London, UK, 2002. Springer-Verlag.

[29] J.-Y. Pan, H.-J. Yang, P. Duygulu, and C. Faloutsos. Automatic image captioning. *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, 3:1987–1990 Vol.3, 27-30 June 2004.

[30] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, pages 1470–1477, October 2003.

[31] J. Tang and P. H. Lewis. A study of quality issues for image auto-annotation with the corel dataset. *IEEE Trans. Circuits Syst. Video Techn.*, 17(3):384–389, 2007.

[32] A. Yavlinsky. *Image indexing and retrieval using automated annotation*. PhD thesis, Imperial College London, Imperial College London, South Kensington Campus, London SW7 2AZ, August 2007.

[33] A. Yavlinsky, E. Schofield, and S. Rüger. Automated Image Annotation Using Global Features and Robust Nonparametric Density Estimation. In W. K. Leow, M. S. Lew, T.-S. Chua, W.-Y. Ma, L. Chaisorn, and E. M. Bakker, editors, *Image and Video Retrieval*, volume 3568 of *LNCS*, pages 507–517, Singapore, 2005. Springer.