On Image Retrieval using Salient Regions with Vector-Spaces and Latent Semantics

Jonathon S. Hare and Paul H. Lewis

Intelligence, Agents, Multimedia Group,
School of Electronics and Computer Science,
University of Southampton,
Southampton, SO17 1BJ,
United Kingdom
{jsh02r, phl}@ecs.soton.ac.uk

Abstract. The vector-space retrieval model and Latent Semantic Indexing approaches to retrieval have been used heavily in the field of text information retrieval over the past years. The use of these approaches in image retrieval, however, has been somewhat limited. In this paper, we present methods for using these techniques in combination with an invariant image representation based on local descriptors of salient regions. The paper also presents an evaluation in which the two techniques are used to find images with similar semantic labels.

1 Introduction

The advantages of salient, or interest, points and regions for image retrieval have been greatly discussed in the literature over the past few years [1, 2]. Previous approaches to retrieval using salient regions have involved directly comparing the local feature descriptors of each region pair in a query and target image. These approaches have then used an algorithm that either sums distance or performs voting, to rank the target images in order of similarity to the query.

In this paper we discuss two approaches inspired by ideas from the field of information retrieval for indexing and retrieving documents:- vector space retrieval models; and Latent Semantic Indexing, or LSI. Both vector space models and LSI have been applied to image retrieval in the past [3, 4, 5, 6, 7], however, with the notable exception of the work of Sivic and Zisserman [3] and previous work by the authors [8], none of the previous works have tried to couple the use of salient regions with these retrieval techniques.

The paper begins by discussing the retrieval techniques and then describes how we coupled salient regions and local descriptors to work with them. The paper concludes with an evaluation of the performance of the two techniques.

2 Information Retrieval Techniques

Recent work by Sivic and Zisserman [3] and slightly earlier work by Westmacott and Lewis [5], showed a new approach to object matching within images and

video footage. The approach was based on an analogy with classical text retrieval using a vector-space model. This section of the paper briefly describes the vector-space model and a second related model of information retrieval called Latent Semantic Indexing or Latent Semantic Analysis.

2.1 Classical Vector-Space Retrieval

Most classical text retrieval systems work in the same general way, by representing a document and query as a set of terms. These terms are represented as axes in a vector-space, using weighted term frequency as the distance along the axis corresponding to that term. Described below are a number of standard steps for this model.

Parsing and Stemming. Firstly, a document is parsed into a list of separate words, this is obviously an easy task in most languages as the words are separated by spaces. The words are then transformed by a process called stemming. The stemming process represents words by their stems, for example, 'CONNECT', 'CONNECTED', and 'CONNECTIONS' are all represented by the stem 'CONNECT'. Words with a common stem will often have similar meanings.

Stop Lists. The next stage is to apply a stop list. The stop list is used to reject common words that occur frequently throughout the corpus of documents, and therefore are not discriminating for a particular document. Examples of such words include words like 'and', 'an' and 'the'.

Representing documents by word frequency. Each of the words from the document (after application of the stop list) are then represented by a unique identifier for that word. The number of occurrences of each word in the document is counted and a vector of word-frequencies created to represent the document.

Frequency weighting. Each component of the vector of word frequencies is often weighted. The standard way of weighting the frequency vectors of text documents is called 'term frequency-inverse document frequency', *tf-idf*, and the default weighting is computed as follows. Suppose that there is a vocabulary of k words, then each document is represented by a k-vector $V_d = (t_1, \ldots, t_i, \ldots, t_k)^T$, of weighted word frequencies with components, $t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$, where n_{id} is the number of occurrences of word i in document d, n_d is the total number of words in the document d, n_i is the number of occurrences of the term i in the whole database and N is the number of documents in the whole database. The weighting is the product of two terms: the *word frequency* n_{id}/n_d and the *inverse document frequency* $\log N/n_i$. The intuition is that word frequency increases the weights of words that occur frequently in a particular document, and thus describe it well, whilst the inverse document frequency down-weights words that appear in many documents in the database. A number of other *tf-idf* weighting functions exist, such as the Okapi BM-25 formula of Robertson *et al* [9] which was found to have superior performance when retrieving text documents.

Indexing using Inverted Files. Inverted file structures are used for efficient retrieval. An inverted file is like an ideal book index. Each word in the collection has an entry in the inverted file, together with a list of documents (and the position at which the word occurs in the document) that contain that word.

Searching: Ranking the results. In order to search the database of documents, a *tf-idf* vector is created for the query terms or document, and the query vector is compared against all the vectors in the database, V_d. The documents in the database are ranked using the normalised scalar product (cosine of angle), $\cos(\theta) = \frac{V_q \bullet V_d}{|V_q||V_d|}$.

2.2 Latent Semantic Indexing

The classical approach to text retrieval described above depends on a lexical match between the words in the query and those in the document collection. However, there is often a lot of diversity in the words used to describe a document (*synonymy*), and the words often have multiple meaning (*polysemy*), making the lexical methods incomplete and imprecise. Deerwester *et al* [10] suggest that it is possible to take advantage of the implicit higher-order structure in the association of terms with documents by determining the singular value decomposition (SVD) of large, sparse, term by document matrices. Terms and documents represented by the k largest singular vectors are then matched against user queries. Deerwester calls this retrieval method Latent Semantic Indexing (LSI) because the k subspace represents important associative relationships between terms and documents that are not evident in individual documents.

The Term-Document Matrix and its Decomposition. LSI begins by constructing a vector space representation for each document, representing each document by a vector of word frequencies, as described in the previous section. The vectors are then arranged into a matrix $\mathbf{A}$, which is known as the term-document matrix. An individual element in $\mathbf{A}$, a_{ij} represents the frequency of term i in document j. The matrix $\mathbf{A}$ is usually very sparse because every word does not normally occur in each document. It is normal to apply weightings to each element of $\mathbf{A}$, such that, $a_{ij} = L(i,j) \times G(i)$, where $L(i,j)$ represents the local weighting for term i in document j and $G(i)$ is the global weighting for term i.

Log-Entropy Weighting. The most commonly used weighting for LSI is the "Log-Entropy" weighting. The local weighting is the log of the term-frequency of an individual document, and the global weighting is related to the entropy of the term frequency over the entire collection. This weighting scheme ensures that a term whose appearance tends to be equally likely among the documents is given a low weight and a term whose appearance is concentrated in a few documents is given a higher weight. The equations for the weighting are as follows, $L(i,j) = \log(tf_{ij} + 1)$, $G(i) = 1 - \sum_{j=1}^{N} \left(\frac{tf_{ij}}{gf_i} \log(\frac{tf_{ij}}{gf_i}) \right) / \log N$, where

tf_{ij} is the frequency of term i in document j, gf_i is the total number of times term i occurs in the entire collection, and N is the total number of documents in the collection.

Decomposition into a subspace. Once the weighted term-document matrix has been created, it is decomposed using the singular value decomposition. Briefly, SVD is used to decompose matrix $\mathbf{A}$ into the product of three separate matrices, $\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V}$, $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$. The monotonically decreasing (in value) diagonal elements of the matrix $\mathbf{\Sigma}$ are called the singular values of the matrix $\mathbf{A}$. These matrices represent the breakdown of the original relationships into linearly-independent vectors or *factor values*. By selecting the first (largest) k singular values of A, it is possible to construct a rank-k approximation to $\mathbf{A}$ via $\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^{\mathrm{T}}$. By reducing the dimensionality of $\mathbf{A}$, much of the "noise" that causes poor retrieval performance is thought to be eliminated.

Queries and Subspace Projection. In order to perform queries in the reduced term-document space, query vectors need to be represented as vectors in the k-dimensional space and compared to each document. Given a query vector, $\mathbf{q}$, whose non-zero elements contain the weighted (using the same weighting as in the creation of the term-document matrix) term-frequency counts of the terms that appear in the query, then, the query vector can be projected into the k-dimensional subspace, $\hat{\mathbf{q}} = \mathbf{q}^{\mathrm{T}}\mathbf{U}_k\mathbf{\Sigma}_k^{-1}$. The k-dimensional query vector, $\hat{\mathbf{q}}$ can then be compared against each of the document vectors and the results ranked. Again, a common similarity measure is the cosine similarity, described in section 2.1.

3 Images as Words

3.1 Salient Regions

Much of the previous work in the field of content-based image retrieval has been based around the concepts of using global descriptors to describe the content of the image. More recently, researchers have begun to realise that global descriptors are not necessarily good enough to describe the actual objects within the images and their associated semantics. Two approaches have grown from this realisation; firstly approaches have been developed whereby the image is segmented into multiple regions, and separate descriptors are built for each region; and secondly, the use of salient points has been suggested.

The first approach has been demonstrated [11], although it has a large problem - that of how to perform the segmentation. Over the years many techniques for performing image segmentation have been suggested, although none really solve the problem of linking the segmented region to the actual object that is being described.

The second approach avoids the problem of segmentation altogether by choosing to describe the image and its contents in a different way. By using salient

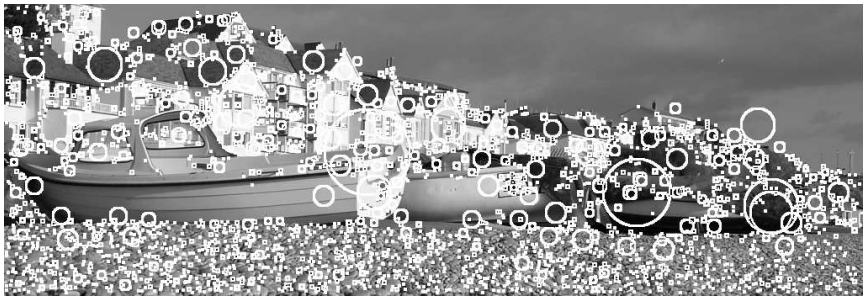

Fig. 1. Example salient regions found from the peaks in the difference-of-Gaussian pyramid

points or regions within an image, it is possible to derive a compact image description based around the local attributes of the salient points.

In previous work, it has been shown that content-based retrieval based on salient interest points and regions performs much better than global image descriptors [1, 2]. For our content-based image retrieval algorithm, we select salient regions using the method described by Lowe [12], where scale-space peaks are detected in a multi-scale difference-of-Gaussian pyramid. Peaks in a difference-of-Gaussian pyramid have been shown to provide the most stable interest regions when compared to a range of other interest point detectors [1, 13]. An example the kinds of salient regions found from the peaks in a difference-of-Gaussian pyramid are shown in Figure 1.

3.2 Local Feature Descriptors

There are a large number of different types of feature descriptors that have been suggested for describing the local image content within a salient region; for example, colour moments and Gabor texture descriptors [2]. The choice of local descriptor is in many respects dependent on the actual application of the retrieval system; for example, some applications may require colour, others may not. In the current implementation of the algorithm, Lowe's SIFT (Scale Invariant Feature Transform) descriptor [12] is used. The SIFT descriptor was shown to be superior to other descriptors found in the literature [14], such as the response of steerable filters or orthogonal filters. The performance of the SIFT descriptor is enhanced because it was designed to be invariant to small shifts in the position of the salient region, as might happen in the presence of imaging noise.

3.3 Creating Visual Terms

One immediately obvious problem with taking local descriptors to represent words is that, depending on the descriptor, there is a possibility that two very similar image patches will have slightly different descriptors, and thus there is a possibility of having an absolutely massive vocabulary of words to describe the

image. A standard way to get around this problem is to apply vector quantisation to the descriptors to quantise them into a known set of descriptors. This known set of descriptors then forms the vocabulary of 'visual' terms that describe the image. This process is essentially the equivalent of the stemming, where the vocabulary consists of all the possible stems. The next problem is that of how to design a vector quantiser. Sivic and Zisserman [3] selected a set of video frames from which to train their vector quantiser, and used the k-means clustering algorithm to find clusters of local descriptors within the training set of frames. The centroids of these clusters then became the 'visual' words representing the entire possible vocabulary. The vector quantiser then proceeded by assigning local descriptors to the closest cluster.

In this work, a similar approach was used. A sample set of images from the data-set was chosen at random, and feature vectors were generated about each salient region in all the training images. Clustering of these feature descriptors was then performed using the batch k-means clustering algorithm with random start points in order to build a vocabulary of 'visual' words. Each image in the entire data-set then had its feature vectors quantised by assigning the feature vector to the closest cluster. On average our test images tended to contain about 3000 salient regions, each represented by a 128-dimensional SIFT key. By transforming the representation into a vector space, each image can be represented by a k-dimensional (or less with LSI) vector of term occurrences.

4 Comparison of the Vector-Space approach with LSI

In order to compare the performance of the vector-space retrieval model and the LSI approach, we have performed an evaluation using a subset of the images from the University of Washington Ground Truth Dataset [15]. We also compare the performance of the two algorithms against a baseline retrieval using a global 64-bin grayscale histogram with images ranked with increasing Euclidean distance between the query images' histogram and the target image. A grayscale histogram has been used as a basis for comparison because the SIFT features are also based on grayscale information and we want to try to avoid any bias that the use of a colour descriptor may contribute.

4.1 Performance Metrics

In order to compare performance, we use two different performance metrics: Semantic Relevance [1] and precision/recall curves. These are described briefly below.

Semantic Relevance. The University of Washington Ground Truth Dataset contains a 697 semantically marked up images. For example an image may have a number of labels describing the image content, such as "trees", "bushes", "clear sky", etc. Given a query image with a set of labels, it is reasonable to expect that the images returned by the retrieval system should have the same labels as

the query image. Let A be the set of all labels from the query image, and B be the set of labels from a returned image. We then define the semantic relevance, $R_{semantic}$, of the query to be:

$$R_{semantic} = \frac{|A \cap B|}{|A|} \quad . \tag{1}$$

This implies that if all the labels in set A exist in set B then the semantic relevance will be 100%, and if only half of the labels in set A exist in set B then the semantic relevance will be 50%.

Precision & Recall: Relevance. In addition to comparing the image retrieval algorithms through the semantic relevance measure, we would also like to plot precision-recall curves. In order to do this, we need to know whether a particular target image is relevant to the query. Using the semantic relevance measure, above, we define the relevance of each image, $V_{n,Z} \in \{0, 1\}$, to be:

$$V_{n,Z} = \begin{cases} 0 \text{ if } R_{semantic} < Z \\ 1 \text{ otherwise} \end{cases} \quad , \tag{2}$$

where Z is a threshold parameter that determines how much semantic relevance a target image must have to be deemed relevant to the query.

4.2 Results and Discussion

We used all 697 semantically marked images from the Washington dataset to form the test set. Each of the images was indexed using the two algorithms, and queries were performed by taking each image in the set as the query image. Vocabulary sizes of 3000, 6000 and 12000 'visual' terms were tested, as well as a range of k values for the LSI technique. Different weighting schemes were also tested. We calculated the semantic relevance measure for the rank 1 image in each query (the closest image, not counting the query), and an averaged semantic relevance over the closest 5 images (rank 1 through 6).

Figure 2 shows the variation of the rank 1 semantic relevance averaged over all queries with respect to the k value for LSI queries with different weightings and a 3000 term vocabulary. The figure shows that optimal retrieval appears to be at a k value of about 47.

Table 1 summarises the performance of the algorithms with respect to their semantic relevance performance. The table shows that LSI-based retrieval (with $k = 47$) outperforms the vector-space method by a small margin, and both methods are much better than retrieval through global grayscale histograms, and certainly much better than random retrieval.

The summarised results in the table also show that the smaller 3000 term vocabulary seems to give the best results. This is an interesting result because the Video Google work of Sivic [3] *et al* used a much larger vocabulary, with a similar algorithm. Further investigation is required in this area to discover what the optimal vocabulary size is.

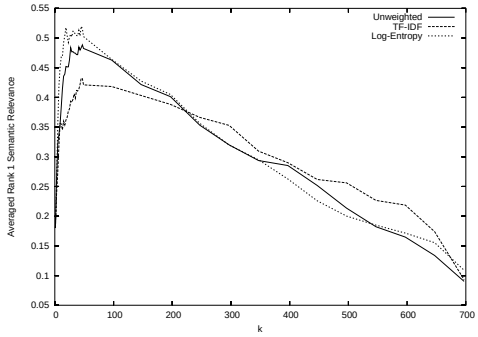

Fig. 2. Effect of varying k with respect to retrieval performance for LSI based retrieval

Figure 3 shows the precision-recall curves of the algorithms with optimal settings (3000 term vocabulary, $k = 47$ with Log-Entropy weighting for LSI, unweighted for vector space), as well as the curves for random retrieval and global grayscale histogram based retrieval.

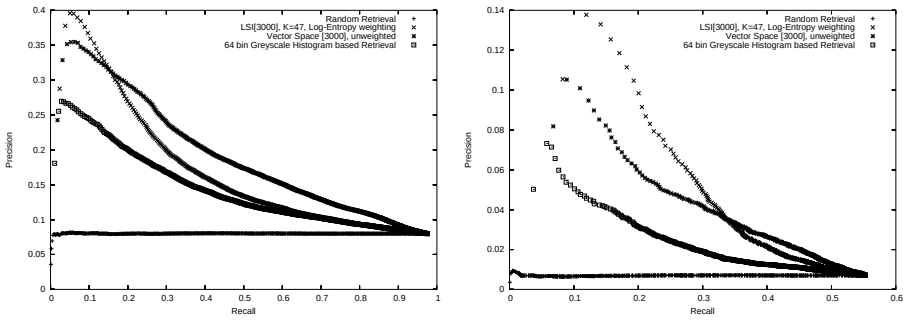

Fig. 3. Precision-Recall plot for Z=0.5 (a) and Z=1.0 (b)

5 Conclusions and Future Work

This paper has presented a way to link methods from the information retrieval community with image description through salient regions to form powerful image retrieval techniques. We have shown how local descriptors from salient regions can be quantised into 'visual' terms and these terms used as a basis for indexing through the vector-space and Latent Semantic Indexing retrieval models.

The evaluation of the two techniques has shown that with well-chosen parameters, the LSI technique exhibits a slightly better performance than the vector-

Table 1. Summary of Retrieval Performance

Method	Weighting	Vocabulary Size	k	Rank 1 Semantic Relevance	Averaged Top 5 Semantic Relevance
LSI	Unweighted	3000	47	0.49	0.38
	TF-IDF	3000	47	0.43	0.35
	Log-Entropy	3000	47	0.52	0.40
LSI	Unweighted	6000	47	0.48	0.38
	TF-IDF	6000	47	0.39	0.33
	Log-Entropy	6000	47	0.50	0.40
LSI	Unweighted	12000	47	0.48	0.39
	TF-IDF	12000	47	0.46	0.39
	Log-Entropy	12000	47	0.50	0.40
Vector Space	Unweighted	3000	N/A	0.45	0.38
	TF-IDF	3000	N/A	0.43	0.36
	Log-Entropy	3000	N/A	0.40	0.34
Vector Space	Unweighted	6000	N/A	0.43	0.37
	TF-IDF	6000	N/A	0.39	0.33
	Log-Entropy	6000	N/A	0.40	0.33
Vector Space	Unweighted	12000	N/A	0.43	0.36
	TF-IDF	12000	N/A	0.38	0.33
	Log-Entropy	12000	N/A	0.38	0.33
64 bin Grayscale Histogram	N/A	N/A	N/A	0.35	0.29
Random Retrieval	N/A	N/A	N/A	0.14	0.14

space technique. Both techniques vastly outperform retrieval by global grayscale histogram matching.

The semantic labels used for marking up the images in the database are in some ways deficient because they use no predefined ontology or vocabulary; for example, some of the images have a "Garbage Can" label, whilst others have a "Trash Can" label. The measure of semantic relevance has no way of knowing these terms have the same meaning. We plan to overhaul the labels describing each of the images by applying a smaller, fixed vocabulary. This should give a better indication of how semantically relevant one image is to another.

Whilst this paper has shown that LSI performs better than the vanilla vector-space model in terms of retrieval performance, no investigation has been performed to look at the computational complexity aspects. The LSI technique requires a certain amount of off-line processing time to construct the SVD of the term-document matrix, but once this is done, queries can be performed in a much reduced dimensional space. We need to investigate how much of an improvement in time to process a query can be gained by this.

Other future work involves investigating the effect of using stop-words on the performance of the vector-space technique. We also plan to incorporate a local colour descriptor to augment the SIFT key descriptors.

6 Acknowledgements

We are grateful to the EPSRC and Motorola UK Research Laboratory for their support of this work.

References

[1] Hare, J.S., Lewis, P.H.: Salient regions for query by image content. In Enser, P., Kompatsiaris, Y., O'Conner, N.E., Smeaton, A.F., Smeulders, A.W.M., eds.: Image and Video Retrieval: Third International Conference, CIVR 2004, Dublin, Ireland, Springer (2004) 317–325
[2] Sebe, N., Tian, Q., Loupias, E., Lew, M., Huang, T.: Evaluation of salient point techniques. Image and Vision Computing **21** (2003) 1087–1095
[3] Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: International Conference on Computer Vision. (2003) 1470–1477
[4] Zhao, R., Grosky, W.I.: From features to semantics: Some preliminary results. In: IEEE International Conference on Multimedia and Expo (II). (2000) 679–682
[5] Westmacott, M., Lewis, P.: An inverted index for image retrieval using colour pair feature terms. In: Proceedings of the SPIE Image and Video Communications and Processing Conference. (2003) 881–889
[6] Squire, D.M., Müller, H., Müller, W.: Improving response time by search pruning in a content-based image retrieval system, using inverted file techniques. In: CBAIVL '99: Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, Washington, DC, USA, IEEE Computer Society (1999) 45
[7] Cascia, M.L., Sethi, S., Sclaroff, S.: Combining textual and visual cues for content-based image retrieval on the world wide web. In: CBAIVL '98: Proceedings of the IEEE Workshop on Content - Based Access of Image and Video Libraries, Washington, DC, USA, IEEE Computer Society (1998) 24
[8] Hare, J.S., Lewis, P.H.: Content-based image retrieval using a mobile device as a novel interface. In Lienhart, R.W., Babaguchi, N., Chang, E.Y., eds.: Proceedings of Storage and Retrieval Methods and Applications for Multimedia 2005, San Jose, California, USA, SPIE (2005) 64–75
[9] Robertson, S.E., Walker, S., Hancock-Beaulieu, M.: Okapi at trec-7: Automatic ad hoc, filtering, vlc and interactive. In: TREC. (1998) 199–210
[10] Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science **41** (1990) 391–407
[11] Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: Image segmentation using expectation-maximization and its application to image querying. IEEE Trans. Pattern Anal. Mach. Intell. **24** (2002) 1026–1038
[12] Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
[13] Mikolajczyk, K.: Detection of local features invariant to affine transformations. PhD thesis, Institut National Polytechnique de Grenoble, France (2002)
[14] Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. In: International Conference on Computer Vision & Pattern Recognition. Volume 2. (2003) 257–263
[15] University of Washington: Ground truth image database. http://www.cs.washington.edu/research/imagedatabase/groundtruth/ (2004)