Jigsaw: Combining a two-phase search approach with an Image Distortion Model for content-based image retrieval

S.S. Evangelatos, D.K. Iakovidis, D.E. Maroulis

Department of Informatics and Telecommunications
University of Athens, Greece
s.evaggelatos@di.uoa.gr

Abstract

In the era of the multimedia-rich World Wide Web and inexpensive digital cameras there is an ever-increasing need for image indexing and retrieval techniques beyond the traditional metadata-based techniques. Content Based Image Retrieval (CBIR) systems facilitate the retrieval of images based on visual content. Jigsaw is a modularized, scriptable and customizable CBIR framework that was built to serve as a research platform. The system extends the query by example paradigm by supporting arbitrarily shaped regions of interest. To the best of our knowledge, Jigsaw is the first system to combine a sliding window indexing technique with the Image Distortion Model similarity measure. To make real-time searches feasible, Jigsaw employs a filtering technique based on sub-image signatures combined with a fast VA-file based multidimensional index. The performance of the system was evaluated, using the Wang image database and simple color features, and was found to be on par with other modern systems, even with the use of very simple color features and without much fine-tuning.

Keywords: CBIR, Content-Based Image Retrieval, Digital Libraries, Image Distortion Model, Multidimensional index

1. Introduction

Content-Based Image Retrieval (CBIR) has been an active field of research in the past few years. CBIR systems, as opposed to text based approaches, enable users to perform queries in image and video databases based on visual content, rather than text-based descriptions and metadata. Even though a variety of systems have been proposed, the problem of CBIR remains difficult and unsolved for many applications.

Below, we present the Jigsaw system, a customisable image retrieval framework building upon the SamMatch retrieval methodology. Jigsaw is the first CBIR system that integrates the two-phase filtering approach of SamMatch [Hua et Al. 1999] [Vu & Tavanapong 2003] with the Image Distortion Model (IDM) [Keysers et Al. 2004]

[Deselaers 2003]. Furthermore, it has the ability to utilise several different image features and uses a fast Vector Approximations file [Weber et Al. 1998] to index the visual information. Using the Jigsaw system, we are able to perform interactive queries, utilizing several different image features.

Our system differs from other common approaches to the CBIR problem in that it does not rely on image segmentation algorithms that often fail to produce a meaningful segmentation. Instead it employs a sliding window approach to index as many as possible sub-images of different sizes from each image. This method allows the system to locate in its database, objects that appear scaled or transposed, compared to the query image. Several other systems have used a sliding window approach such as the wavelet based Warlus [Natsev et al. 2004] system or the SamMatch [Vu & Tavanapong 2003] [Hua et al. 1999] system. One of the challenges that systems like these face is the vast number of sub-image signatures that need to be indexed. Previous efforts to address this problem have been proposed that rely on clustering techniques combined with an R-tree based index [Vu & Tavanapong 2003]. In the Jigsaw system, a VA-file based index has been used that is able to handle interactive queries without the extra step of clustering.

Furthermore, we employ the IDM (Image Distortion Model) similarity model for ranking the results. The IDM has the benefit of being robust in slight deformations or misalignments of the compared images. This property of the IDM allows us to use a larger sliding window step resulting in a lower number of sub-images.

From a software engineering perspective, the system focuses on modularity, extensibility and being scriptable in order to support automated experiments. It has to be stressed that Jigsaw, being a modular system, can be reconfigured and extended to support different retrieval strategies. Below, we present our experiments and conclusions from the initial configuration of the system.

2. Basic Concepts

2.1 Noise Free Queries

The most common query model supported by current CBIR systems is "Query by Example" (QBE), according to which the CBIR system searches an image database for images that are similar to an example image presented to the system. The concept of Noise Free Queries (NFQ) [Vu & Tavanapong 2003] is an extension to the QBE model that enables the user to define a free-form Region Of Interrest (ROI) on the example image, thus excluding non relevant pixels of the image from the query. In Jigsaw, the region of interest is defined by a grayscale mask. The values of the mask for each pixel represent the relative significance of the pixel to the query.

2.2 A two phase approach

In every interactive retrieval system there are two, often conflicting, requirements that have to be met, response time and accuracy. Most image similarity algorithms require a significant amount of computing time to access the similarity of two images. Thus it is infeasible to directly compare the similarity of the query image to every image in the database. Instead, in the Jigsaw framework a multidimensional vector (image signature) is computed for each sub-image of an image and a multidimensional index is used to retrieve the most relevant signatures from the database. Afterwards, the sub-images that the relevant signatures came from are ranked according to their relevance to the query using a more accurate, but time consuming, similarity model.

3. Indexing Methodology

3.1 Image features

The first step in the process of indexing an image is feature extraction. During feature extraction, the image is divided in $n \times n$ non-overlapping pixel blocks and a feature vector is extracted from each block. Each feature vector can include a combination of several available image features. Depending on the application, a combination of the following features can be used: a) the average intensity b)the mean hue c) the mean color in several color spaces (RGB, CIE-Lab or HSV) d) the mean chromaticity values e) a fuzzy grayscale histogram [Deselaers 2003] f) any combination of the Tamura texture features [Tamura et Al. 1978] (contrast, coarseness and directionality) g) a vector of the n most significant colors. The feature extraction process may be repeated twice to extract two sets of features. One set of features is used to compute the sub-image signatures and a, possibly different, set of features may be stored to be later used in the similarity ranking phase.

3.2 Indexing with a Sliding Window

While the feature vectors are being extracted, they are placed on a grid, according to the location on the image that they originated from. We call this grid, FVM (Feature Vector Map). In the next step, square windows of different sizes are used to group the features vectors that originated from the corresponding sub-image. This group of feature vectors takes part in the computation of a sub-image signature as it is described in the following section. Then the indexing window slides a few samples and a different signature is computed until the whole area of the image is indexed. The process is repeated with indexing windows of different sizes ranging in size from windows that cover most of the image at once, to windows small enough to isolate single objects.

3.3 Sub-image signatures

From each sub-image, an image signature is extracted that captures the spatial distribution of image features in the sub-image. Each signature is multidimensional vector of real values. For the signature computation the indexing window is divided in seven overlapping regions (fig. 1). Those are the following: a) the whole area of the indexing window b) the two diagonals of the indexing window and c) the four disjoint quadrants of the indexing window. For each of the seven regions, a statistical average-variance pair (μ,σ) per feature vector dimension is computed. The final form of the signature vector coming from n-dimensional feature vectors is shown in eq. 1.

$$\underbrace{(\underbrace{\mu_1^1,\sigma_1^1,\ldots,\mu_n^1,\sigma_n^1}_{2n},\underbrace{\mu_1^2,\sigma_1^2,\ldots,\mu_n^2,\sigma_n^2}_{2n},\ldots\underbrace{\mu_1^7,\sigma_1^7,\ldots,\mu_n^7,\sigma_n^7}_{2n})}_{14n} \qquad (1)$$

Signatures have the property of being invariant to scaling transformations of the sub-image while being able to capture the spatial distribution of the features on the sub-image. The downside of the signatures methodology is the resulting high dimensionality of the signature vectors as the resulting dimensionality is 14 times the dimension of the feature vectors.

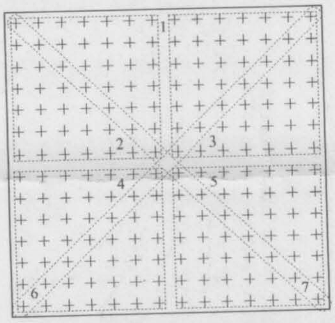

Figure 1. The seven overlapping signature regions

3.4 The VA-file index

The high dimensionality of the signature vectors makes the use of a typical R-tree based index prohibitively slow. It has been shown that the performance of such space partitioning indexes degrades rapidly as the dimensionality increases [Weber et Al. 1998]. Instead a VA-file based multidimensional index is used. A VA-file index takes advantage of the high linear read speed of the modern hard disks and the caching mechanisms provided by the OS by performing a linear scan in a table containing highly space efficient approximations of the actual vectors. To form these space efficient approximations, the signature vectors are quantized and stored in a binary file along with pointers to the actual vectors.

4. Retrieval Methodology

4.1 Query signature

In order to query the index for similar signatures, a signature has to be computed for the query. The first step in computing the signature is extracting features from the ROI of the query. Then a signature must be computed for the ROI from the extracted features. Unfortunately the signatures, as defined so far, can only be computed for square image regions while the ROI of the query can take any shape. To overcome this problem, a square area has to be located that contains as much of the ROI as possible while keeping the amount of non-ROI pixels low. We call this square area the "Core Area". The core area is located by an algorithm that starts by assuming that the core area covers the entire image and iteratively shrinks it until the average interest value of the pixels in the area is higher than 80% and it is not possible to further shrink the area without leaving out more ROI pixels. Finally, the feature vectors of the ROI that lie inside the core area are used to compute the query signature as described in section 3.3.

4.2 Filtering by querying the signatures index

The query signature is finally used to perform a k-nearest neighbour search among the signature vectors in the VA-file index. The Manhattan distance is used during the k-nn search to speed up the procedure. The number of the signatures retrieved is a trade-off between speed and accuracy and it is specified by the user along with the query. The more signatures are returned from the index, the more expensive it gets to rank the results in the next stage. On the other hand, if too few signatures are returned, the recall and precision of the system are reduced since fewer images will be available in the ranking phase, where the more accurate similarity evaluation takes place.

4.3 Ranking results

The final stage of a query is similarity ranking. Each of the signatures retrieved from the index in the previous step refers to a sub-image in the image database. We refer to those sub-images as candidate sub-images. In order to compare a candidate sub-image with the query image, the core area of the query is aligned with each one of the candidate sub-images and the Image Distortion Model algorithm is applied to evaluate the similarity of the query ROI with the area around the candidate sub-image and assign a similarity score to the candidate sub-image. It is crucial to point out that the IDM comparison is not constrained in the sub-image but extends in a region around the sub-image similarly shaped to the query ROI.

For the IDM to perform the comparison, image features have to be available for both images. The database sub-image features can be extracted on the fly or loaded from disk if they were extracted offline (section 3.1). The query features are extracted on an as needed basis right before the similarity evaluation. Since the size of the sub-images and the query can vary, the density of the extracted query features is adapted to have an equal number of feature vectors in the candidate sub-image and the core area. The query features are organized on a feature vector map and cached in order to be used in subsequent comparisons.

The IDM computes a similarity score by performing, for each feature vector in the ROI of the query, a local search for the best matching feature vector among the vectors of a small neighborhood in the corresponding area of the database image. The distances of the matching vectors area aggregated in a similarity score value. The procedure is depicted in figure 2. The local search performed from the IDM compensates for slight misalignments, scale, viewing angle or shape variations between the compared images.

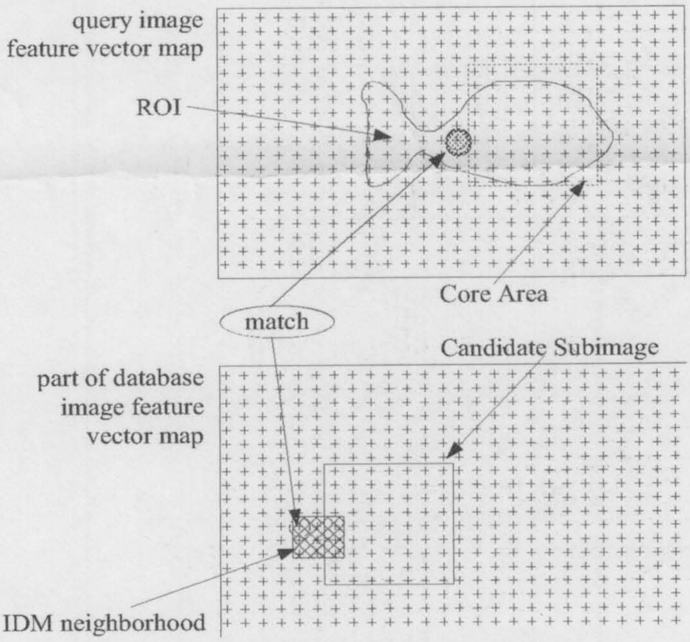

Figure 2. *The seven overlapping signature regions*

The computation of the distance between a query and a database image using the IDM is expressed by the formula in equation 2 where Q is a query and B is a database image, with feature vectors q_{ij} and b_{ij} respectively while w_{ij} are weight values that are derived from the ROI mask and reflect the interest values that the user has assigned in different areas of the query. The sum term is used to normalize the results according to the number of feature vectors used.

$$D_{IDM}(Q,B) = \frac{\sum_{ij} w_{ij} \min_{i'=i-r}^{i+r} \min_{j'=j-r}^{j+r} D(q_{ij}, b_{i'j'})}{\sum_{ij} 1} \qquad \textbf{(2)}$$

The similarity distance value is converted to a similarity score by composition with a monotonically decreasing function.

Finally, each matching image is assigned the highest among the similarity scores of the sub-images that it contains. The matching images are presented to the user, through a web interface, in descending similarity score order. In figure 3 the top ranking results of a sample query can be seen.

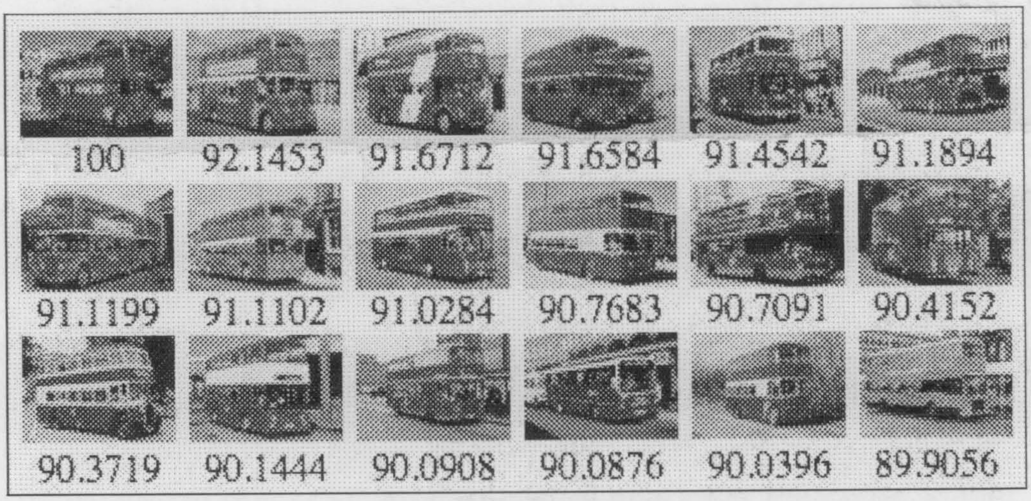

Figure 3. *A sample query in the Wang database. The upper left picture was used as the query.*

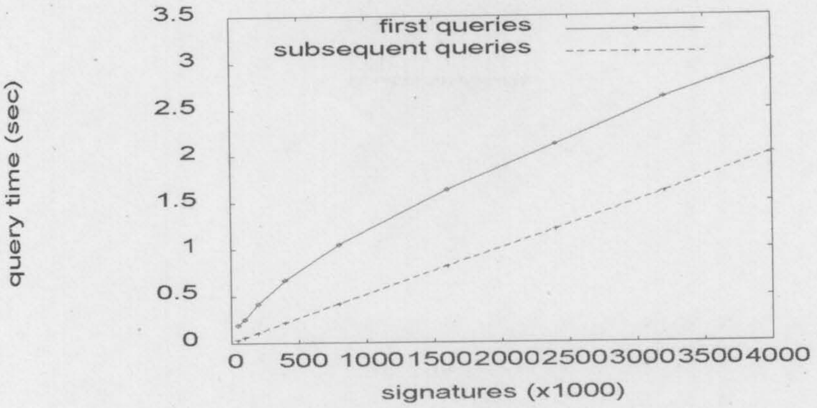

Figure 4. Scaling of the VA-file index

5. Experimental Study

5.1 Index Performance

The performance of the multidimensional index is crucial to the interactive usage of the system. The sliding window indexing technique produces a vast amount of signature vectors that need to be indexed. The usage of a VA-file index allowed us to avoid clustering the signature vectors, thus improving the precision of the system. The query times of the index scale linearly to the number and the dimensionality of the signature vectors (fig. 4 & 5). In each of the experiments, the 10^3 nearest neighbours to a query vector were retrieved.

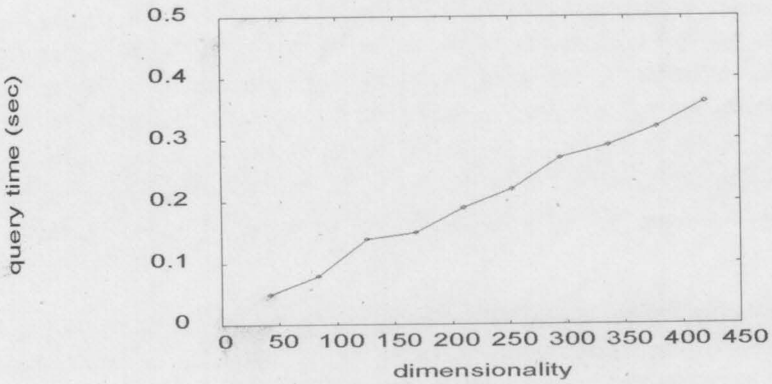

Figure 5. Scaling of the VA-file index as a function of signature dimensionality

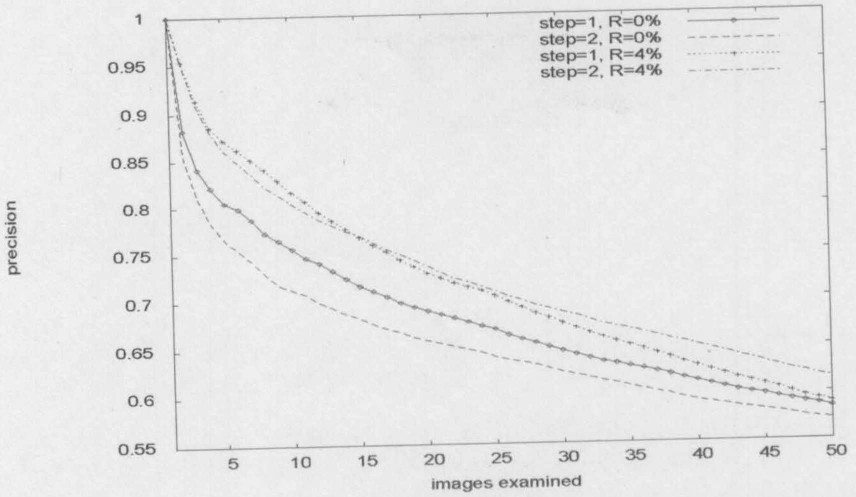

Figure 6. *Precision* (relevant images/images examined) *of the system*

5.2 Query Results

To assess performance of the system, the free Wang [Wang et Al. 2001] image database was used. The Wang database consists of 1000 images evenly distributed in 10 semantically different classes such as beach settings, busses, dinosaurs, flowers, elephants, horses, mountain settings etc. All images were normalized to 384×256 pixels.

Ten images were randomly selected from each class to be used as queries, using the whole image as the ROI. The database was populated using all 1000 images. As it was not our intention to measure the effectiveness of different image features, we used only the average CIE-Lab colour as a feature both for the signatures and the similarity ranking. Feature vectors were extracted from 8×8 pixel blocks and indexing windows of three different scales were used (256×256, 192×192 and 128×128 pixels). The configuration above resulted in 42-dimensional signature vectors.

In each of the queries, the 5000 most similar sub-images were retrieved from the index. The step of the indexing sliding window was varied between one and two blocks and the precision of the system was measured taking into account the first 50 images. By alternating the value of the radius of the search neighbourhood in the IDM to 0% and 4% of the sub-image size, we were able to measure the influence of the distortion in the precision of the system. We verified that the distortion has a significant influence in the precision of the system and enables it to compensate for