WEB IMAGE INDEXING: COMBINING IMAGE ANALYSIS WITH TEXT PROCESSING

Luís A. Alexandre[*], *Manuela Pereira, Sara C. Madeira, João Cordeiro and Gaël Dias*

ArtIC - Artificial Intelligence and Computation group
Computer Science Department - University of Beira Interior - Covilhã, Portugal
[*]Networks and Multimedia Group, IT, Covilhã, Portugal
{lfbaa, mpereira, smadeira, jpaulo, ddg}@di.ubi.pt

ABSTRACT

In this paper we describe a web image indexing and retrieval system called ARTISTIC that allows text and/or image queries. Unlike other systems that only process the text in HTML tags, in the image caption or in the page title, ARTISTIC processes the complete page text and uses keywords (relevant terms with eventually more than one word) to index the images. Traditional color and texture features are also used.

1. INTRODUCTION

MPEG-7 sets a standard for multimedia description in order to efficiently and effectively describe and retrieve multimedia information [1]. However, finding useful descriptors is difficult as they have to be searched in an eclectic environment and seldom implies cognitive issues. In order to tackle these problems, we propose a methodology that combines textual information and image features in order to describe the contents of images in a search engine framework. There are several systems to search for images on the web, that use text information: WebSeer [2], WebSeek [3], the system described in [4] and WebMARS [5]. There are also the image versions of the mainstream search engines, such as, Alltheweb, Altavista, Ditto, Excite, Google, Lycos and Picsearch. Among these, only Google seems to process the text page beyond the image file names and HTML tags (although it is not easy to know for sure since the details are not made public). These systems suffer from one or more of the following drawbacks: the text in the web page is only partially processed; only simple words are considered as textual features; it is not clear how textual information is used to support image indexing and retrieval; term lists or taxonomies are built in the setup phase of the system with user intervention; directory-to-term conversion tables have to be created by hand. ARTISTIC has a clear algorithm for using the complete page text information to aid image indexing; it is a non-supervised system (no user interaction is needed for setup); it is language independent; it supports both image and text queries; it uses multiword units (See Section 4) and not just single words as keywords.

Section 2 introduces the general scheme of ARTISTIC . Section 3 and 4 respectively present image analysis and text processing details. The process of text and image queries is explained in section 6.

2. GENERAL SCHEME

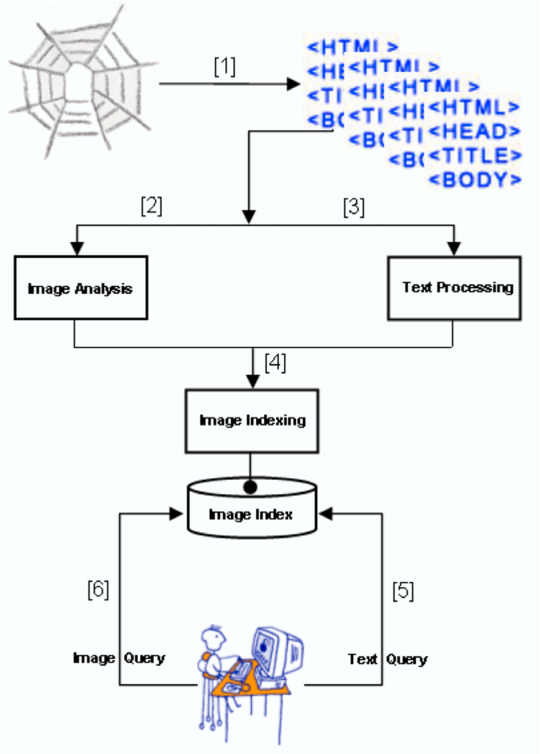

Fig. 1. General scheme

The general scheme of ARTISTIC is divided into six main steps (see figure 1). First, a softbot gathers all the web pages of a given site in the web. Second, the page images are extracted and their characteristics are processed. In parallel, the useful textual information in the web pages is extracted (step 3). Finally, the image indexing process is carried out (step 4). The user can now perform image and/or text queries based on the computed image index (steps 5-6).

3. IMAGE ANALYSIS

ARTISTIC is able to read JPEG, GIF and PNG images. These account for the majority of image file types in the web. We use information from color and texture to characterize an image. The analysis is done on seven predefined regions (which include the image as a whole too). These regions are the white portions in figure 2. Note that the use of regions conveys spatial information, making the color features yield color layout information. The information from the color and texture is combined into a 840-D feature vector to represent each image in the feature space.

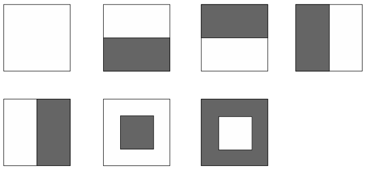

Fig. 2. 7 image regions used to determine image features.

3.1. Color features

Color features are the most commonly used features to characterize images in the context of image retrieval. They are independent of image size and orientation and are relatively robust to background noise [6]. Among the possible features, color histograms are preferred since they yield a good representation of the color distribution in a compact form.

To extract the color features, the image is transformed from RGB to HSV color space. This color space has a color representation closer to human perception than RGB. The first set of features are color histograms: three color histograms (one for each color component) with 32 bins each are calculated for each of the seven regions. The choice of 32 bins represents a compromise between a sparse histogram (one with many bins, which has high noise sensibility) and one with poor representation capability (with few bins). The color histograms are normalized, such that the sum of the values for all bins of each color component sum to one. The color histogram values are included in the vector feature representation of the image. The fact that this information is included in the vector feature representation solves the problem of the combination of similarity measures from different approaches. The second set of features are color moments: the first and second moments are found for each of the seven regions and for each color component, thus resulting in 42 features.

3.2. Texture features using DWF

Theoretical and implementation aspects of wavelet based algorithms in texture characterization are well studied and understood. Following Mallat's initial proposal [7], many researchers have examined the utility of various wavelet representations in texture analysis [8, 9, 10]. Unser's experiments [9] suggest that filters play an important role in texture description. In wavelet approaches, texture is usually characterized by its energy distribution in the decomposed subbands. Simple norm-based distances, together with heuristic normalization are also used. However, in [11] the authors show that the modeling of marginal distribution of wavelet coefficients using the generalized Gaussian density (GGD) and a closed form of the Kullback-Leibler distance between GGDs provide great accuracy and flexibility in capturing texture information.

In the present work, we employ the discrete wavelet frames (DWF) using the 9-7 biorthogonal filter [12] that present in [13] better results than the 8-tap Daubechie orthogonal wavelets proposed in [11]. Given an image, the DWF decomposes it using the same method as the wavelet transform, but without the subsampling process. This results in four filtered images with the same size as the input image. The decomposition is then continued in the LL channels only as in the wavelet transform, but since the image is not sub-sampled, the filter has to be up-sampled by inserting zeros in between its coefficients. The main advantages of the wavelet frame representation are that it focuses on scale and orientation texture features, it decomposes the image into orthogonal components and it is translation invariant. So, we then use the method proposed in [11] that we briefly expose. The GGD, is defined as:

$$p(x;\alpha,\beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|x|/\alpha)^{\beta}}, \qquad (1)$$

where $\Gamma(.)$ is the Gamma function, i.e. the following expression $\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt,\ z > 0$. Here, α models the width of the PDF peak (variance), while β is inversely proportional with the decreasing rate of the peak. Given the GGD model, the PDF of the wavelet coefficients at a subband can be completely specified by the two parameters α and β. The closed form of the Kullback-Leibler distance (KLD) between two GGDs is:

$$D(p(x;\alpha_1,\beta_1)||p(x;\alpha_2,\beta_2)) = log\left(\frac{\beta_1\alpha_2\Gamma(\beta_2)}{\beta_2\alpha_1\Gamma/\beta_1)}\right) + \left(\frac{\alpha_1}{\alpha_2}\right)^{\beta_2} \frac{\Gamma((\beta_2+1)/\beta_1)}{\Gamma(1/\beta_1)} - \frac{1}{\beta_1}. \qquad (2)$$

Using the chain rule of KLD [14] with the reasonable assumption that wavelet coefficients in different subbands are independent, the overall similarity between two images is the sum of the KLDs given in equation (2) between corresponding pairs of subbands. The method used yield 2 features per wavelet subband. We use three scales of decomposition, thus we have 9 subbands. Using the regions presented in figure 2, we have a total of $2 \times 9 \times 7 = 126$ features per image.

4. TEXT PROCESSING

Extracting useful information from texts is a crucial issue in Information Retrieval, and especially in Multimedia Information Retrieval. In particular, two kinds of information should be evidenced: information about the language (i.e. multiword units) and information about the text content (i.e. keywords).

On one side, extracting multiword units (MWUs) from texts is the first step towards text normalization. MWUs include a large range of linguistic phenomena, such as phrasal verbs (e.g. "to go for the ball"), nominal compounds (e.g. "free kick") and named entities (e.g. "Manchester United"). MWUs are frequently used in everyday language, usually to precisely express ideas that cannot be compressed into a single word. Therefore, it is clear that their identification is crucial for language understanding and consequently for correct text indexing. For this purpose, multiword units are extracted from the available web pages using a statistically-based software called SENTA (Software for the Extraction of N-ary Textual Associations) [15]. SENTA is particularly suitable for our task since it is language independent enabling its application to any page on the web without predefining language heuristics.

On the other side, the indexing task can be considered as the identification of a set of keywords that defines the text content. In the context of our work, we define a keyword as a relevant word or a pertinent multiword unit. In order to correctly index texts, we use a well-known methodology introduced by G. Salton [16] called the $tf.idf$ score. This score is defined in equation 3 where t is a term (a word or a MWU) and p is a web page.

$$tf.idf(t,p) = \frac{tf(t,p)}{|p|} \times log_2 \frac{N}{df(t)} \tag{3}$$

For each t in p, we compute the term frequency $tf(t,p)$ that is the number of occurrences of t in p and divide it by the number of terms in p, $|p|$. We then compute the inverse document frequency of t by taking the log_2 of the ratio of N, the number of web pages in our experiment, to the web page frequency of t, that is the number of web pages in which the term t occurs ($df(t)$). As a result, a term occurring in all web pages will have an inverse document frequency 0 giving him no chance to be a keyword. A term which occurs very often in one web page but in very few web pages of the collection will have a high inverse document frequency thus a high $tf.idf$ score. Consequently, it will be a strong candidate for being a keyword.

The text processing ends with a list of words and multiword units associated with their $tf.idf$ score. These data will be filtered out in the next step of our architecture: the image indexing process.

5. IMAGE INDEXING

Image Indexing can be defined as the process that associates a set of keywords to an image thus defining its content. For this purpose, we propose an innovative unsupervised methodology based on the textual information that surrounds the image.

First, we associate to each image the set of all the terms that are in the same web page or in the web page that the image refers to[1]. This can be viewed as the following expression:

$$\forall i_k \in I, i_k \mapsto \{t_{k1}, ..., t_{kn}\} \tag{4}$$

where t_{kj} is any term in the set of all terms T related to i_k, which is any image in the set of all images I.

Since not all the terms are good keywords, the best ones need to be selected. As a consequence, the next step aims at evaluating the relationship between each term and the image. For that purpose, it is clear that terms evidencing a high $tf.idf$ score should be preferred. However, the proximity between the term and the image must also be taken into account. It is obvious that the more distant a term is from the image, the less it should be considered as a potential keyword. Thus we introduce a straight forward relation between a term t_{kj} and the image i_k:

$$dti(t_{kj}, i_k) = \frac{1}{|pos(t_{kj}, i_k)|} \tag{5}$$

where $dti(t_{kj}, i_k)$ is the *term-image distance* and $pos(t_{kj}, i_k)$ is the number of terms that separates the first occurrence of the term t_{kj} from its corresponding image i_k. It is important to notice that $pos(t_{kj}, i_k)$ is negative when the term precedes the image and positive when it follows it.

After the second step the reader can easily conclude that a term with a high $tf.idf$ score and a high dti is a strong keyword candidate. However, this assumption can be strengthened. Indeed a term which is highly concentrated aside the image should be preferred to those terms that spread along the text. For that purpose, we introduce a new measure of density [2]:

$$dens(t_{kj}) = \sum_{q=1}^{Q-1} \frac{1}{dist(occur(t_{kj}, q), occur(t_{kj}, q+1))} \tag{6}$$

[1] In the latter case, it is more probable that the referred text deals with the topic of the image.

[2] Our measure follows the idea of [17].

where $dens(t_{kj})$ is the density of the term t_{kj}, Q is the number of occurrences of the term t_{kj} in the text and the expression $occur(t_{kj}, q)$ denotes the q^{th} occurrence of t_{kj}.

To conclude, a good indexing term should evidence a high $tf.idf$ score, a high dti and a high density. This assumption is supported by the following *relevance measure*:

$$\begin{aligned} weight(t_{kj}, i_k) \ = \ tf.idf(t_{kj}, p_{i_k}) \ & \times \ dti(t_{kj}, i_k) \\ & \times \ dens(t_{kj}) \end{aligned} \tag{7}$$

where $weight(t_{kj}, i_k)$ is the relevance function and the following expression $tf.idf(t_{kj}, p_{i_k})$ is the $tf.idf$ score of the term t_{kj} in the web page text p_{i_k} that contains image i_k[3].

Once all the terms related to a given image have been evaluated the selection process must be carried out. This task aims at choosing the best keyword candidates. For that purpose, a term is chosen as keyword candidate if its relevance measure exceeds the average term-image $weight(.,.)$ by some threshold number of standard deviations. For instance, all terms in $\{t_{k1}, ..., t_{kn}\}$ exceeding the average by two standard deviations should be selected as keywords to index the i_k image.

6. QUERY AND RETRIEVAL

When text is used to perform a query, ARTISTIC searches in the image index for images that are associated with the query. The images are ranked according to their similarity score.

An image can also be used to perform a query. The 840-D feature representation of the query image is obtained. The closest[4] images in the feature space are analyzed and their keyword lists are combined. This list is then used to expand the query. The final output is a ranked list of images (1) ordered according to their similarity with the query image, (2) ordered according to their similarity computed using the keywords that expand the query.

7. CONCLUSIONS

In this paper, we propose a web image indexing and retrieval system, ARTISTIC that allows text and/or image queries. The interest of combining information from both text and images in a Multimedia search engine is obvious. Unlike most systems that do not take into account the complete textual information, ARTISTIC proposes an innovative unsupervised approach that combines full textual information with image characteristics (such as color and texture) for accurate image indexing and retrieval.

[3] It is obvious that all three measures are normalized in order to give equivalent weight to each one

[4] The notion of closeness is defined by a statistical measure similar to the one used in section 5 for keyword selection.

8. REFERENCES

[1] J. Martínez, R. Koenen, and F.Pereira, "Mpeg-7: the generic multimedia content description standard," *IEEE Multimedia*, vol. 9, no. 2, 2002.

[2] M. Swain, C. Frankel, and V. Athitsos, "Webseer: An image search engine for the world wide web," in *CVPR*, 1997.

[3] J. Smith and S. Chang, "An image and video search engine for the world-wide web," in *Storage. Retr. Im. Vid. Datab.*, pp. 84–95. SPIE, 1997.

[4] G. Amato, F. Rabitti, and P. Savino, "Multimedia document search on the web," in *7th Int. WWW Conf.*, Brisbane, Australia, Ap. 1998.

[5] M. Ortega-Binderberger, S. Mehrotra, K. Chakrabarti, and K. Porkaew, "Webmars: A multimedia search engine," in *SPIE An. Sym. Elect. Im.*, San Jose, California, Jan. 2000.

[6] Y. Rui, T. Huang, and S. Chang, "Image retrieval: past, present and future," in *Int. Sym. Mult. Inf. Proc.*, Taiwan, Dec. 1997.

[7] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pat. An. Mach. Int.*, vol. 11, pp. 674–693, July 1989.

[8] T. Chang and C. Kuo, "Texture analysis and classification with tree structured wavelet transform," *IEEE Trans. Im. Proc.*, vol. 2, pp. 429–441, Oct. 1993.

[9] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Im. Proc.*, vol. 4, pp. 1549–1560, Nov. 1995.

[10] A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Trans. Pat. An. Mach. Int.*, vol. 15, pp. 1186–1191, Nov. 1993.

[11] M. Do and M. Vetterli, "Texture similarity measurement using kullback-leibler distance on wavelet subbands," in *IEEE ICIP*, Vancouver, Canada, Sept. 2000.

[12] M. Antonini, M.Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Im. Proc.*, vol. 1, pp. 205–230, Ap. 1992.

[13] A. Mojsilovic, M. Popovic, and D. Rackov, "On the selection of an optimal wavelet basis for texture characterization," *IEEE Trans. Im. Proc.*, vol. 9, no. 12, pp. 2043–2050, Dec. 2000.

[14] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley Interscience, New York, NY, 1991.

[15] G. Dias, S. Guilloré, and J. Lopes, "Mining textual associations in text corpora," in *6th ACM SIGKDD Work. Text Mining*, Boston, USA, Aug. 2000.

[16] G. Salton and C. Buckley, "Global text matching for information retrieval," *Science*, vol. 253, pp. 1012–1025, 1991.

[17] B. Jun-Peng, S. Jun-Yi, L. Xiao-Dong, L. Hai-Yan, and Z. Xiau-Di, "Document copy detection based on kernel method," in *Int. conf. nat. lang. proc. knowl. eng.*, Beijing, China, Oct. 2003.