Toward Cross-Language and Cross-Media Image Retrieval

Carmen Alvarez, Ahmed Id Oumohmed, Max Mignotte, Jian-Yun Nie
DIRO, University of Montreal
CP. 6128, succursale Centre-ville
Montreal, Quebec
H3C 3J7 Canada
{bissettc, idoumoha, mignotte, nie}@iro.umontreal.ca

Abstract

This report describes the approach used in our participation of ImageCLEF. Our focus is on image retrieval using text, i.e. Cross-Media IR. To do this, we first determine the strong relationships between keywords and types of visual features. Then the subset of images retrieved by text retrieval is used as examples to match other images according to the most important types of features of the query words.

1 Introduction

The RALI group at University of Montreal has participated in several CLEF experiments on Cross-Language IR (CLIR). Our participation in this year's ImageCLEF experiments is to see how our approach can be extended to Cross-Media IR. Two research groups are involved in this task: one on image processing and the other on text retrieval. Our CLIR approach is similar to that used in our previous participation in CLEF, i.e. we use statistical translation models trained on parallel web pages for French to English translations. For the translation from other languages, we use bilingual dictionaries. Our focus is on image retrieval from text queries.

Different approaches have been used for image retrieval. 1) A user can submit a text query, and the system can search for images using image captions. 2) A user can submit an image query (using an example image - either selected from a database or drawn by the user). In this case, the system tries to determine the most similar images to the example image by comparing various visual features such as shape, texture, or color. 3) There is still a third group of approaches which tries to assign some semantic meaning to images. This approach is often used to annotate images by concepts or by keywords [1]. Once images have been associated with different keywords, they can be retrieved for a textual query.

The above three approaches have their own advantages and weaknesses.

The first approach is indeed text retrieval. There is no particular image processing. The coverage of the retrieval is limited to images with captions.

The second approach does not require the images to be associated with captions. However, the user is required to provide an example image and a visual feature or a combination of some features to be used for image comparison. This is often difficult for a non-expert user.

The third approach, if successful, would allow us to automatically recognize the semantics of images, thus allow users to query images by keywords. However, the development up to now only allows us to annotate images according to some typical components or features. For example, according to a texture analysis, one can recognize a region of image as corresponding to a tiger because of the particular texture of tigers [2]. It is still impossible to recognize all the semantic meanings of images.

Some recent studies [3] have tried to automatically create associations between visual features and keywords. The basic idea is to use a set of annotated images as a set of learning examples, and to extract strong associations between annotation keywords and the visual features of the images. In our study, we initially tried to use a similar approach in ImageCLEF. That is, we wanted to extract strong relationships between the keywords in the captions and the visual features of the images. If such relationships could be created, then it would be possible to use them to retrieve non-annotated images by a textual query. In this case, the relationships play a role of translation between media. However, we discovered that this approach is extremely difficult in the context of ImageCLEF for several reasons:

1. The annotations (captions) of the images in the ImageCLEF corpus often contain keywords that are not strongly associated with particular visual features. They correspond to abstract concepts. Examples of such keywords are "Scotland", "north", and "tournament". Therefore, if we use the approach systematically, there will be many noisy relationships.

2. Even if there are some relationships between keywords and visual features, these relationships may be difficult to be extracted because there are a huge number of possible visual features. In fact, visual features are continuous. Even if we use some discretization techniques, their number is still too high to be associated with some keywords. For example, for a set of images associated with the keyword "water", one would expect to extract strong relationships between the keyword and the color and texture features. However, "water" in images may only take up a small region of the image. There may be various other objects in the same images, making it difficult to automatically isolate the typical features for "water".

Due to these reasons, we take a more flexible approach. We also use the images with captions as a set of training examples, but we do not try to create relationships between keywords and particular visual features (such as a particular shade of blue for the word "water"). We only try to determine which type(s) of feature is (are) the most important for a keyword. For example, "water" may be associated with "texture" and "color". Only strong relationships are retained. During the retrieval process, a text query is first matched with a set of images using image captions. This is a text retrieval step. Then the retrieved images are used as examples to retrieve other images, which are similar according to the determined types of features associated with the query keywords. The whole processes of our system is illustrated in figure 1.

In the following sections, we will first describe the image processing developed in this project. In particular, we will describe the way that relationships between keywords and visual features are extracted, as well as image retrieval with example images. In section 3, we will describe the CLIR approach used. In section 4, both approaches are combined to perform image retrieval. Section 5 will describe the experimental results and some conclusions.

Our approach is much less ambitious than that of [3], but it is more feasible in practice. In fact, in many cases, image captions contain abstract keywords that cannot be strongly associated with visual features, and even if they can, it is impossible to associate a single vector to a keyword. Our approach does not require determining such a single feature vector for a given keyword. It abandons the third approach mentioned earlier, but combines the first two families of approaches. The advantage of extracting keyword-feature associations is to avoid the burden of requiring the user to indicate the appropriate types of features to be used in image comparison.

2 Image processing-based learning procedure

The objective of the automatic image processing-based learning procedure that we propose in this section is twofold:

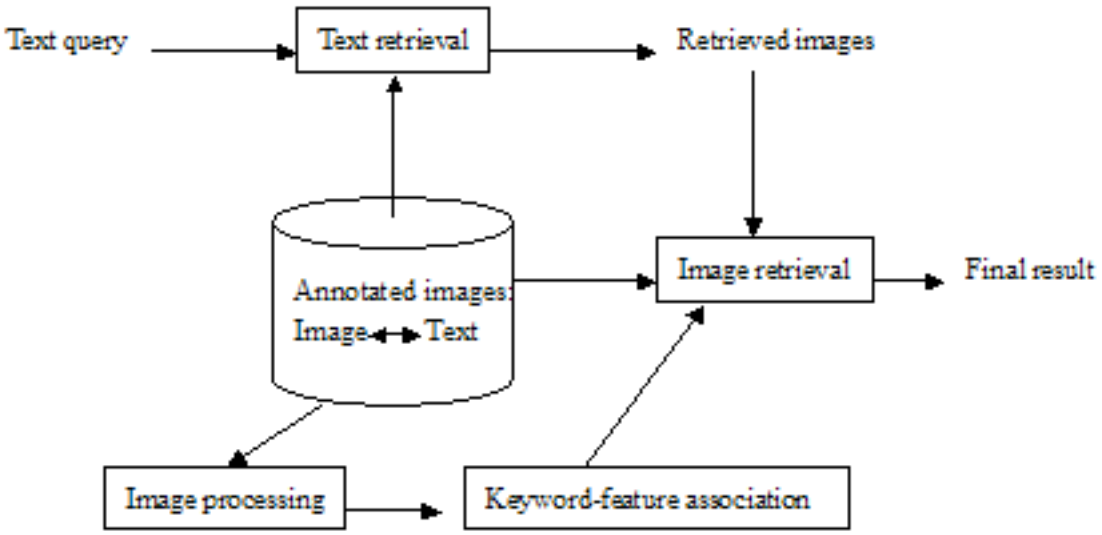


Figure 1: Workflow of image retrieval

- First, this procedure aims at estimating the most discriminant type(s) of high-level visual features for each annotated keyword. In our application, we have considered the three fundamental visual characteristics; namely, *texture* (including color information), *edge* and *shape*. For example, the keyword "animal" could belong to the *shape* class since the measure using *shape* information will be the most discriminant to identify images with animals (but the more specific keywords "zebra" and "tiger" will more probably belong to the *edge* and *texture* classes respectively due to the characteristic coat of these animals).
  A discriminant measure, belonging to each of these classes of visual features has then been defined. We have considered :

  1. The mean and the standard deviation of the energy distribution in each of the sub-band of an Harr wavelet [4] decomposition of the image as discriminant measure of the *edge* class.
  2. The coarseness measure proposed by Tamura *et al.* [5] as discriminant measure of the *texture* class.
  3. The histogram of the edge orientation of the different shapes extracted from the input image (after a region-based segmentation) as discriminant measure of the *shape* class.

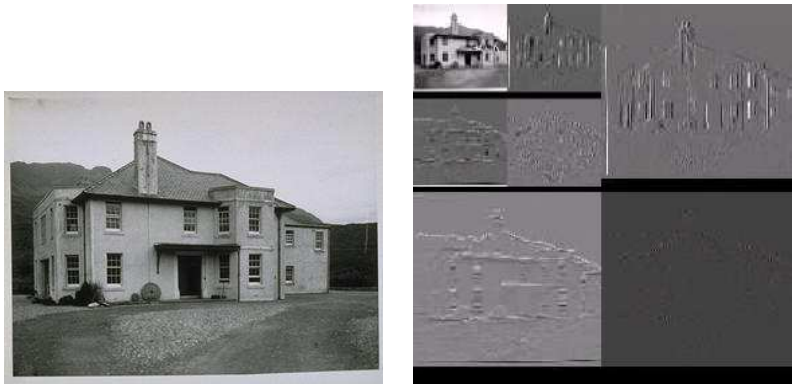
- The second objective is to identify a set of candidate images that are the most representative for each annotated keyword, in the sense of similarity distance combining one or several pre-estimated visual feature classes.

The type of high-level visual feature (along with its discriminant measure) and a set of candidate images along with its associated normalized similarity distance will be used with cross-language Information, to refine the retrieval process.

2.1 Edge class and its measure

Wavelet-based measures have often been used in content-based image retrieval system because of the appealing ability to describe the local texture and the distribution of the edges of a given image at multiple scales. In our application we use the Harr wavelet transform [4] for the luminance (i.e., grey-level) component of the image. There are several other wavelet transforms but the Harr wavelet transform has better localization properties and requires less computation compared to

other wavelets (e.g., Daubechies' wavelet). The procedure of image decomposition into wavelets involves recursive numeric filtering. It is applied to the set of pixels of the digital image which is decomposed with a family of orthogonal basis functions obtained through translation and dilatation of a special function called *mother* wavelet. At each scale (or step) in the recursion, we obtain four sub-bands (or sets of wavelet coefficients), which we refer to as LL, LH, HL and HH according to their frequency characteristics (L : Low and H : High, see Figure 2). The LL sub-band is then decomposed into four sub-bands at the next scale decomposition. For each scale decomposition (three considered in our application), we compute the mean and the standard deviation of the energy distribution (i.e., the average and the square of each set of wavelet coefficients) in each of the sub-bands. This leads to a vector of 20 (i.e., $(2 \times 3 \times 3) + 2$) components or attributes which can be viewed as the descriptor (or the signature) of the *edge* information/characteristics of the image. For example, an image containing a zebra thus has high energy in the HL sub-band and low energy in the LH sub-band due to the vertical strips of the coat of this animal.



(**a**) (**b**)

Figure 2: *a) The original image* STAND03_1028/STAND03_26737_BIG.JPG *of size* 238×308. *b)The Haar wavelets coefficients after the image a) is adjusted to size* 256×256.

2.2 Texture class and its measure

Tamura *et al.* [5] have proposed to characterize image texture along the dimensions of contrast, directionality, coarseness, line-likeness, regularity and roughness. These properties correspond to the human texture perception.

– Contrast is a scalar value related to the amount of local intensity variations present in an image and involves the standard deviation of the grey-level probability distribution.

– Directionality is a global texture property which is computed from the oriented edge histogram, obtained by an edge detector like the Sobel detector [6]. The directionality measures the sharpness of the peaks in this edge histogram.

– In this class of visual features, we have utilized only the coarseness property which yields a histogram of six bins, for the following reasons :

- The contrast is not very discriminant for textural retrieval,
- The edge information has been already treated in the wavelet and shape class,
- The line-likeness, regularity and roughness properties are correlated to the coarseness, contrast and directionality properties.

Coarseness refers to the size of the *texton*; i.e., the smallest unit of a texture. This measure thus depends on the resolution of the texture. With this measure, we can compute a histogram

with 6 bins (i.e., a 6-component attribute vector) which will be used as the descriptor of the *texture* characteristics of a given image. The procedure for computing the coarseness histogram is outlined below,

1. At each pixel with coordinates (x, y) in the image, and for each value k (k taking its value in $\{1, 2, \ldots, 6\}$), we compute the average over its neighborhood of size $2^k \times 2^k$ i.e.,

$$A_k(x,y) = \sum_{i=(x-2^{k-1})}^{i=(x+2^{k-1}-1)} \sum_{j=(y-2^{k-1})}^{j=(y+2^{k-1}-1)} \frac{I(i,j)}{2^{2k}}$$

   where $I(i, j)$ is the intensity pixel of the image at pixel (i, j).

2. At each pixel, and for the horizontal and vertical directions, we compute the differences between pairs of averages corresponding to pairs of non-overlapping neighborhoods just on opposite sides of the pixel. The horizontal and vertical differences are expressed as:

$$\begin{aligned} E_{k,horizontal} &= |A_k(x + 2^{k-1}, y) - A_k(x - 2^{k-1}, y)| \\ E_{k,vertical} &= |A_k(y + 2^{k-1}, y) - A_k(y - 2^{k-1}, y)| \end{aligned}$$

3. At each pixel, the value of k that maximizes $E_k(i, j)$, in either direction (horizontal or vertical), is used to set the best size $S_{best}(i, j) = 2^k$. At this stage we can consider, as descriptor, the scalar measure of coarseness which is the average of S_{best} over the entire image, or consider, as in our application, the histogram (i.e., the empirical probability distribution) of S_{best} which is more precise for discrimination.

2.3 Shape class and its measure

Description and interpretation of shapes contained in an input image remains a difficult task. Several methods use a contour detection of the images (such as the Canny or Sobel edge detectors) as a preliminary step in the shape extraction. But these methods remain dependent on certain parameters as thresholds (on the magnitude of the image gradient).

In image compression, some approaches [7] use a vector quantization method on the set of vectors of dimension K^2 of grey-levels corresponding to $K \times K$ blocks extracted from the image. By using a clustering procedure into K classes, we can obtain an image with separate regions (a set of connected pixels belonging to a same class) from which we extract the contours of the different regions. These edges are connected and obtained without any parameter adjustment and the noise is taken into consideration in this procedure. Figure 3 shows an example of edge detection using three regions, i.e., three clusters in the vector quantization.

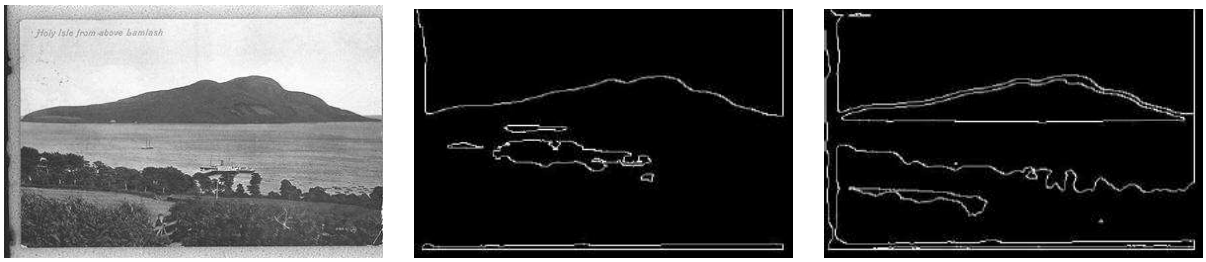
In our application, we use this strategy of edge detection and we use, as clustering procedure, the Generalized LLoyd [8] [9] (generally used in this context). In our implementation, we use the code provided from the QccPack Library[1].

For each edge pixel, we define a direction (horizontal, vertical, first or second diagonal) depending on the disposition of its neighboring edge pixels. For each direction we count the number of edge pixels associated with it, which yields a 4 bin histogram.

2.4 The learning procedure

Given a training database, we first pre-compute and store *off-line* for each image its three descriptors (related to each of the three visual features). These set of three vectors simplify the

[1] http://qccpack.sourceforge.net/



(**a**) (**b**) (**c**)

Figure 3: *a) The original image* STAND03_2093/STAND03_7363_BIG.JPG *b)* 4×4 *pixel blocks and clustering result into* 2 *regions c)* 4×4 *pixel blocks and clustering result into* 3 *regions*

representation of each image, by giving maximal information about its content (according to each considered feature). We now define a similarity measure between two images given a visual feature class. This measure is simply the euclidean distance between two vectors.

The learning procedure which allows us to determine the type of high-level visual feature (and its measure) that is the most representative for each annotated keyword, is outlined below:

1. Let $\mathbf{I}_w$ be the set of all images I_w (each described by its three vectors or descriptors $[D_{I_w}^{texture}, D_{I_w}^{edge}, D_{I_w}^{Shape}]$) in the training database that are annotated with the keyword w and $|\mathbf{I}_w|$, the number of images in $\mathbf{I}_w$.
2. For each CLASS { *Texture*, *Edge*, *Shape* }
   (a) We use a K-mean clustering procedure [6] (with a euclidean distance for the similarity measure) on the set of samples $D_{I_w}^{class}$.
   (b) This clustering allows us to approximate the distribution of the set of samples $D_{I_w}^{class}$ by K spherical distributions (with identical radius) and to give K prototype vectors $[D_{1,w}^{class}, ..., D_{k,w}^{class}]$ corresponding to the centers of these distributions. Several values of K are used to find the best clustering representation of $D_{I_w}^{class}$.
      i. For each PROTOTYPE VECTOR { $D_{1,w}^{class}, ..., D_{k,w}^{class}$ }
         - We search in the whole training database for the closest descriptors (or images) of $D_{k,w}^{class}$, according to the euclidean distance. Let $\mathbf{I}_{k,w}^{class}$ be this set of images.
         - We compute the number of the first top-level T samples of $\mathbf{I}_{k,w}^{class}$ also belonging to $\mathbf{I}_w$ (best results are obtained with $T = 10$). Let $N_{k,w}^{class}$ be this number.
3. We retain the CLASS(ES) and $\mathbf{I}_{k,w}^{class}$ for which we have $N_{k,w}^{class}$ above a given threshold ξ.
4. We normalize in $[0, 1]$ all the similarity distances of each sample of each selected set $\mathbf{I}_{k,w}^{class}$.
5. We combine the similarity distance measures of the selected sets $\mathbf{I}_{k,w}^{class}$, with an identical weighting factor, in order to find a final set of images i associated with each annotated keyword w. The similarity measures of these final images are then normalized, and the noramlized similarity measure of an image i for the given word w is represented as $R_{cluster}(i, w)$ for retrieval, described in section 4.

The first 24 images of the set of images associated to the word *garden* are shown in Figure 4. We can see that, even if most images are not annotated by the word *garden* (the word does not exist in any field of the text associated with the image), we can visually count about 9 images which are related to gardens from the 14 non-annotated images.

Figure 4: *Result of learning procedure applied to the word "garden". Below each image, we can read its identifier key in the database and the score (similarity measure) obtained after normalization. The images which are not annotated by the word "garden" have their identifier key written in a gray box.*

3 Cross-language text retrieval

3.1 Translation models

Two approaches are used for query translation, depending on the ressources available for the different languages. For Spanish, Italian, German, Dutch, Swedish, and Finnish, FreeLang bilingual dictionaries [2] are used in a word-for-word translation approach. The foreign language words in the dictionaries are stemmed using Snowball stemmers[3], and the English words are left in their original form. The queries are also stemmed, and stop words are removed with a stoplist in the foreign language. The translated query consists of the set of all possible English word translations for each query term, each translated word having equal weight.

For French, a translation model trained on a web-aligned corpus is used [10]. The model associates a list of English words and their corresponding probabilities with a French word. As

[2] http://www.freelang.net
[3] http://snowball.tartarus.org

with the bilingual dictionaries, the French words are stemmed, and the English words are not. Word-for-word translation is done. For a given French root, all possible English translations are added to the translated query. The translation probabilities determine the weight of the word in the translated query. The term weights are represented implicitly by repeating a given translated word a number of times according to its translation probability. For French as well as for the other languages, the words in the translated query are stemmed using the Porter stemming algorithm.

This query translation approach was found to be optimal, using training data described in the following section. The parameters evaluated were:

- Whether to use a bilingual dictionary, or the translation model, for French.
- For a given query term, whether to pick just one translation from the dictionary or translation model, all translations, or in the case of the translation model, the first n probable translations.
- When to stem the English words: The English words could be stemmed in the dictionary, rather than after translation. This affects the number times a particular word appears, and therefore its implicit weight, in the final translated query. Without stemming English words in the dictionary, multiple forms of a word may appear as a possible translation for a foreign language stem. After the translated query is stemmed, the English root appears several times.

3.2 CLIR process

For retrieval, the Okapi retreival algorithm [11] is used, implemented by the Lemur Toolkit for Language Modeling and Information Retrieval. In particular, the BM 25 weighting function is used. The following parameters contribute to the relevance score of a document (an image annoatation) for a query:

- BM 25 k1
- BM 25 b
- BM 25 k3
- FeedbackDocCount: the number of documents (image captions) to use for relevance feedback
- FeedbackTermCount: the number of terms to add to the expanded query
- qtf: the weight of the query terms added during relevance feedback

The training data used to optimize each of these parameters, as well as the translation approaches described in section 3.1 was the TREC-6 AP89 document collection and 53 queries in English, French, Spanish, German, Italian, and Dutch. Since no training data was available for Finnish and Swedish, the average of the optimal values found for the other languages is used.

While the training collection, consisting of news articles about 200-400 words in length, is quite different from the test collection of image captions, the volume of the training data (163000 documents, 25 or 53 queries, depending on the language, and 9403 relevance assessments) is much greater than the training data provided from the image collection (5 queries, 167 relevance assessments).

Once the parameters for relevance feedback and the BM 25 weighting function are optimized with the training data, retreival is performed on the test data, producing a list of images and their relevance scores, for each query. We annotate this image relevance score for a query, based on textual retrieval, as $R_{text}(i, q)$.

4 Combining text and images in image retrieval

4.1 The image relevance score based on clustering

The image analysis based on clustering, described in section 2.4, provides a list of relevant images i for a given word w, with a relevance score for each image, $R_{cluster}(i, w)$. The relevance score of an image for a query, based on clustering, is then a weighted sum of the relevance scores for that image for each query term:

$$R_{cluster}(i, q) = \sum_{w \in q} \lambda_w R_{cluster}(i, w) \tag{1}$$

In our approach, each word has the same weight, and the relevance score for the query is normalized, with $\lambda_w = \frac{1}{|q|}$, where $|q|$ is the number of words in the query.

4.2 Combining the five image relevance scores

We now have 5 lists of images for each query, with the following relevance scores:

- $R_{text}(i, q)$
- $R_{cluster}(i, q)$
- $R_{edge}(i, q)$: The similarity between the query image q and a collection image i, according to the wavelet measure described in section 2.1.
- $R_{texture}(i, q)$: The similarty according to the texture class measure from section 2.2.
- $R_{shape}(i, q)$: The similarity according to the shape class measure from section 2.3.

Each of these relevance scores contributes to a final relevance score as follows :

$$\begin{aligned} R(i, q) &= \lambda_{text} R_{text}(i, q) + \lambda_{cluster} R_{cluster}(i, q) \\ &+ \lambda_{edge} R_{edge}(i, q) + \lambda_{texture} R_{texture}(i, q) + \lambda_{shape} R_{shape}(i, q) \end{aligned} \tag{2}$$

The coefficients we chose for the contribution of each approach are as follows:

- $\lambda_{text} = 0.8$
- $\lambda_{cluster} = 0.1$
- $\lambda_{edge} = \lambda_{texture} = \lambda_{shape} = 0.033$

These values have been determined empirically using the training data provided in ImageCLEF.

4.3 Filtering the list of images based on location, photographer, and date

A final filtering is applied to the list of images for a given query. A "dictionary" of locations is extracted from the location field in the entire collection's annotations. Similarly, a "dictionary" of photographers is extracted. If a query contains a term in the location dictionary, then the location of a potential image, if it is known, must match this location. Otherwise, the image is removed from the list. The same approach is applied to the photographer. Similarly, if a date is specified in the query, then the date of the image, if it is known, must satisfy this date constraint.

5 Experimental results and conclusion

A preliminary analysis shows that our image retrieval works well. In particular, using the French queries, our system produced the best results among the participants. This may be related to two factors: - The method of query translation used for these queries is reasonable. For French queries, we used a statistical translation model trained on parallel web pages. This translation model has produced good results in our previous CLIR experiments.

- The method based on keyword-feature type association we used in these experiments may be effective. However, further analysis has to be carried out to confirm this.

For the experiments with other languages, our results are relatively good - they are often among the top results. However, the absolute MAP is lower than for the French queries.

References

[1] W. Liu, S. Dumais, Y. Sun, H. Zhang, M. Czerwinski, and B. Field. Semi-automatic image annotation, 2001.

[2] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, and Jitendra Malik. Blobworld: A system for region-based image indexing and retrieval. In *Third International Conference on Visual Information Systems*. Springer, 1999.

[3] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *ACM SIGIR*, 2003.

[4] S.G. Mallat. A theory for multiresolution signal decomposition : The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.

[5] H. Tamura, S. Mori, , and T. Yamawaki. Texture features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8:460–473, 1978.

[6] S. Banks. *Signal processing image processing and pattern recognition*. Prentice Hall, 1990.

[7] M. Goldberg, P. Boucher, and S. Shlien. Image compression using adaptative vector quantization. *Communications, IEEE Transactions on [legacy, pre - 1988]*, 34:180–187, 1986.

[8] S.P. Lloyd. Last square quantization in pcm's. *Bell Telephone Laboratories Paper*, 1957.

[9] Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, COM-28:84–95, 1980.

[10] Jian-Yun Nie and Michel Simard. Using statistical translation models for bilingual ir. In *Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, pages 137–150, 2001.

[11] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Proc. of the Third Text REtrieval Conference (TREC-3), NIST Special Publication 500-225*, 1995.