Automatic word assignment to images based on image division and vector quantization

Yasuhide Mori, Hironobu Takahashi and Ryuichi Oka

Real World Computing Partnership, Information Basis Function Laboratory,
Tsukuba Mitsui Building 13F, 1-6-1 Takezono, Tsukuba-shi, Ibaraki 305-0032, JAPAN
E-mail: {ymori,hironobu,oka}@rwcp.or.jp

Abstract

We propose a method that relates images and words. This method is based on statistical learning from image databases with words. The method uses two processes. The first uniformly divides each image into sub-images. With this division, all words assigned to images are inherited by each sub-image. The second process clusters sub-images by vector quantization. These processes produce results which show that each sub-image can be correlated to a set of words, each of which is selected from words assigned to original images. After clustering, the voting probability of each word for a set of divided images is estimated. This is done for each cluster of the feature vector of sub-images. Experiments show that this method is effective.

1 Introduction

To make information completely accessible over the Internet, media-independent access methods must be developed. For example, an image is needed as a possible query to retrieve images (Flickner et al., 1995) and text.

To retrieve text queried by an image, text information is automatically extracted from an image at some stage in the retrieval process. Various approaches regarding word annotation for images have been investigated (Kurita et al., 1992; Picard & Minka, 1995, Ono et al., 1996; Watanabe & Nagao, 1998). However, these approaches are very limited in terms of vocabulary or the domain of images. For real data, it is impossible to segment objects in advance, to assume the number of categories, or to avoid the presence of noise which is difficult to erase.

In this paper, we describe a method for image-to-word transformation based on statistical learning. This method uses images with words attached for learning. The method's key procedure can be explained as follows: (1) each image is divided into many parts, and at the same time, all words attached to each image are inherited by each part; (2) parts from all of the images are clustered in a feature space through vector quantization; (3) the likelihood of each word in each cluster is estimated statistically.

The following sections describe the procedure used in the proposed method, give experimental results, discuss our approach, and reveal our conclusions.

2 Procedure of the proposed method

2.1 Motivation and outline

To find the detailed correlation between text and image (not simply discriminating an image into a few categories), each portion of the image should be correlated to words instead of the whole image to words.

Assigning keywords to images portion by portion would be an ideal way to prepare learning data. However, with the exception of a very small vocabulary, we cannot find such learning data nor can we prepare them. The more the size of the data increases, the more difficult assigning keywords to images portion by portion becomes. So we have to develop an another method to avoid this fundamental problem.

To avoid this problem, we propose a simple method to correlate each portion of an image to key words only using key words for the whole image.

The procedure of the proposed method is as follows :

1. Many images with key words are used for learning data,
2. Divide each image into parts and extract features from each part,
3. Each divided part inherits all words from its original image,
4. Make clusters from all divided images using vector quantization,
5. Accumulate the frequencies of words of all partial images in each cluster, and calculate the likelihood for every word,
6. For an unknown image, divide it into parts, extract their features, and find the nearest clusters for all divided parts. Combine the likelihoods of their clusters, and determine which words are most plausible.

The main point of this method is to reduce noise (i.e. unsuitable correlating) by accumulating similar partial patterns from many images with key words. For example, suppose an image has two words, 'sky' and 'mountain'. After dividing the image, the part which has only the sky pattern also has 'sky' and 'mountain' due to the inheriting of all words. The word 'mountain' is inappropriate for the part. However if an another image has two words, 'sky' and 'river', accumulating these two images, the sky pattern has two 'sky"s, one 'mountain' and one 'river'. In such way, we can hope that the rate of inappropriate words are gradually decreased by accumulating similar patterns. [1]

Figure 1 shows the concept of estimating likelihoods of data.

2.2 Dividing image, feature extraction, and inheriting key words

Each image is divided equally into rectangular parts because it is the simplest and fastest way to divide images. The number of divisions ranges from 3×3 to 7×7. In this paper, the dividing method driven by the contents of images such as region extraction has not been tried.

In parallel with the dividing, all words given for an image are inherited into each of the divided parts. This is a straightforward way to give words to each part because there is no informations to select words at this stage.

Extracted features for the divided images are (1) a 4×4×4 cubic RGB color histogram and (2) an 8-directions × 4-resolutions histogram of intensity after Sobel filtering, which can be calculated by fast and common operations.

Feature (1) is calculated as follows:

1. divide RGB color space into 4×4×4 cells,
2. count the number of pixels fall in each cell.

As a result, 64 features are calculated.

Feature (2) is calculated as follows for 4-resolutions (1, 1/2, 1/4 and 1/8) respectively:

1. filtering by vertical (S_y) and horizontal (S_x) Sobel filters,
2. for each pixel, calculate arguments ($\tan^{-1}(S_y/S_x)$),
3. divide arguments $[-\pi, \pi)$ into 8 directions,
4. sum the intensity ($\sqrt{S_x^2 + S_y^2}$) of each pixel in each direction.

As a result, 32 features are calculated.

As a result of these operations, a total of 96 features are calculated from a divided image.

[1] Moreover, we hope that these inappropriate words may convey the correlation between two different kinds of patterns in a database.

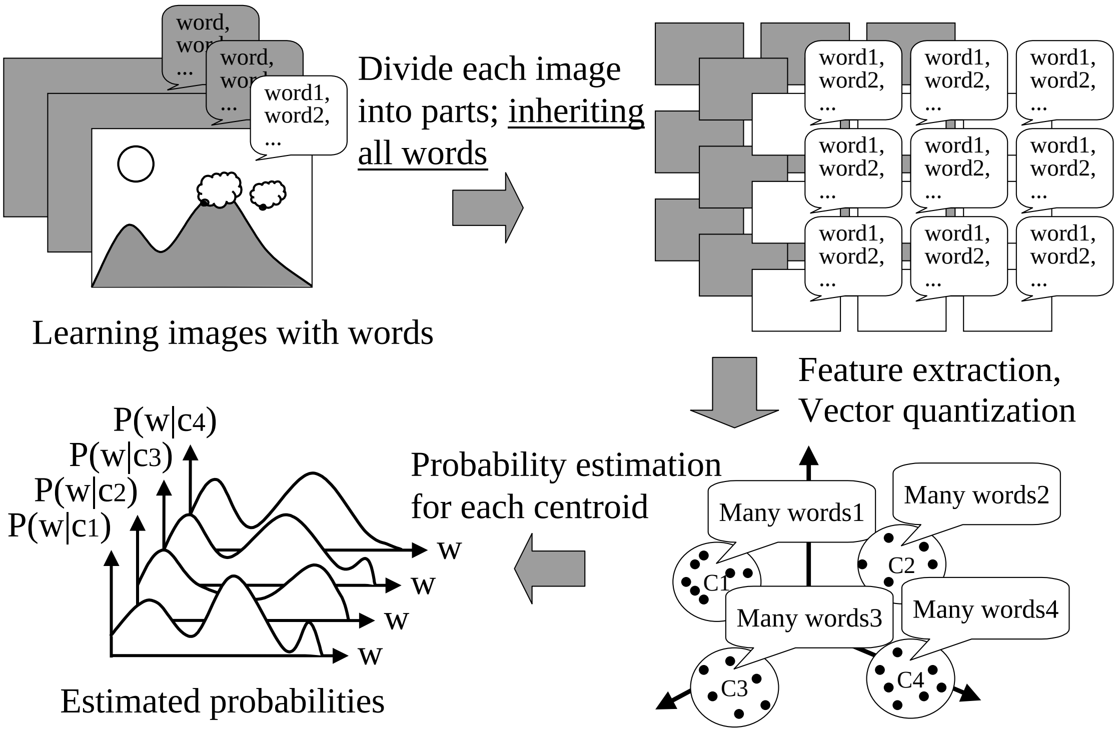

word,
word,
word1,
word2,
...
Divide each image into parts; inheriting all words
Learning images with words
Feature extraction, Vector quantization
Probability estimation for each centroid
Many words1
Many words2
Many words3
Many words4
C1
C2
C3
C4
P(w|c4)
P(w|c3)
P(w|c2)
P(w|c1)
w
Estimated probabilities

Figure 1: Concept of the proposed method.

2.3 Vector quantization

The feature vectors extracted from the divided parts of all learning images are clustered by vector quantization in a 96-dimensional space. In this paper, data incremental vector quantization is used. In this method centroids (representative vectors for each cluster) are created incrementally for data input. Each cluster has one centroid and each data belongs to a cluster uniquely.

There is only one control parameter in this method, that is, the threshold of error for quantization (referred to later as *scale*). The less a scale is, the more centroids are created.

The procedure for vector quantization is as follows:

1. Set the scale d,
2. Select a feature vector as the first centroid,
3. **For** the feature vector of the i-th data ($2 \leq i \leq$ *number of data*):
   if there are centroids such that the distance[2] from the i-th feature is less than d,
   then, the i-th feature vector belongs to the nearest centroid,
   else, set the i-th feature vector as a new centroid.

2.4 Probability estimation of key words for each cluster

After centroids c_j ($j = 1, 2, ..., C$) are created by the vector quantization, likelihoods (conditional probability) $P(w_i|c_j)$ for each word w_i ($j = 1, 2, ..., W$) and each c_j are estimated by accumulating their frequency:

$$P(w_i|c_j) = \frac{P(c_j|w_i)P(w_i)}{\sum_{k=1}^{W} P(c_j|w_k)P(w_k)} \approx \frac{(m_{ji}/n_i)(n_i/N)}{\sum_{k=1}^{W}(m_{jk}/n_k)(n_k/N)} = \frac{m_{ji}}{\sum_{k=1}^{W} m_{jk}} = \frac{m_{ji}}{M_j}, \qquad (1)$$

[2] Euclid distance in the feature space

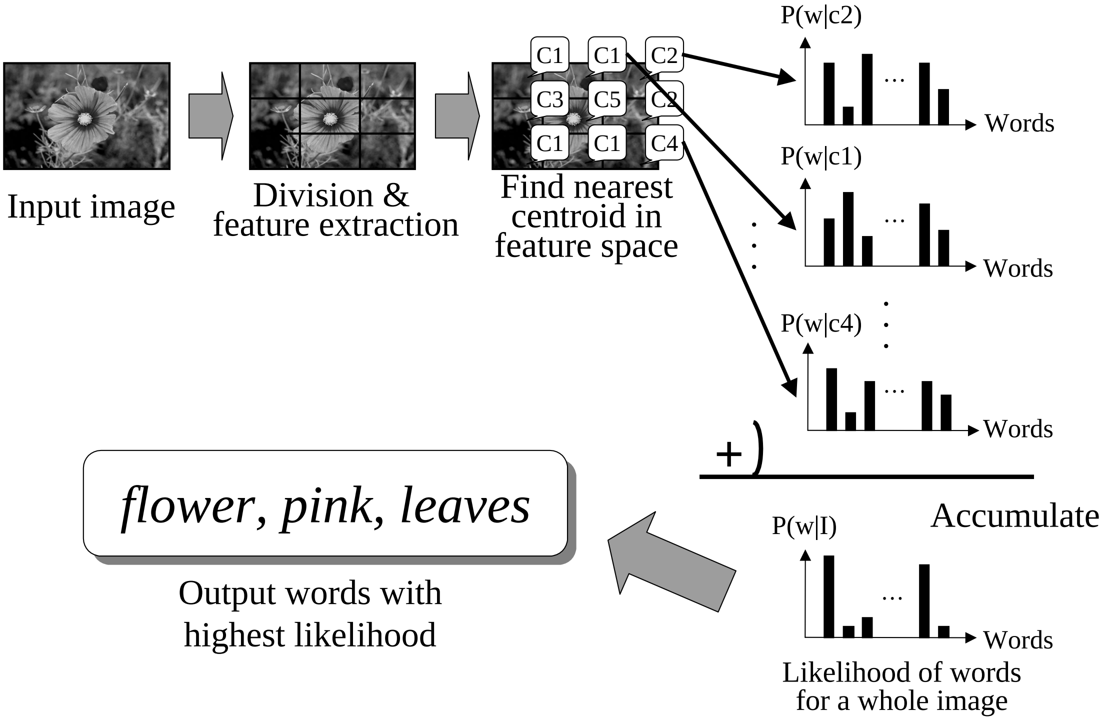

Input image
Division & feature extraction
Find nearest centroid in feature space
C1
C1
C2
C3
C5
C2
C1
C1
C4
P(w|c2)
P(w|c1)
P(w|c4)
Words
Words
Words
+)
Accumulate
P(w|I)
Words
Likelihood of words for a whole image
flower, pink, leaves
Output words with highest likelihood

Figure 2: Concept of determining correlated words from an unknown image.

where, m_{ji} is the total of word w_i in centroid c_j, $M_j (= \sum_{k=1}^{W} m_{jk})$ means the total of all words in centroid c_j, n_i is the total of word w_i in all data, and $N (= \sum_{k=1}^{W} n_k)$ is the total of words for all data (each word is counted repeatedly each time it appears).

2.5 Determining correlated words from an unknown image

Using estimated likelihood $P(w_i|c_j)$, correlated words are determined for an unknown image as follows: First, an unknown image is divided into parts and its features are extracted in the same way as for the learning data. Second, the nearest centroids are found for all divided parts in the feature space. Third, an average of the likelihoods of the nearest centroids is made. Then, words which have the largest average value of the likelihoods are output.

Figure 2 shows the concept of determining correlated words from an unknown image.

3 Experiment and results

3.1 Data preparation

In the experiment, a multimedia encyclopedia[3] is used as an original database. The encyclopedia contains about 60,000 items and about 10,000 images in total. About 10,000 items which have citations to images are selected from all items. Therefore, the data for the experiment consists of about 10,000 pairs of images and corresponding documents (in Japanese).

The accurate number of images used for the experiment are 9,681. There are various kinds of images; landscapes, architecture, historical materials, plants (photographs and sketches), portraits, paintings, etc. About 80% of the images are in color. They have 256 grades of

[3] *Mypaedia*, Hitachi Digital Heibonsha, 1998

brightness and their sizes average 400×280 pixels.

On the other hand, a set of words is extracted from the documents using the following procedure:

1. Divide documents in all items into words [4] and determine each word's part-of-speech (noun, verb, adjective, etc.) using a morpheme analysis program called "*Chasen*" (resulting in about 100,000 vocabularies),
2. Select only (common and proper) nouns and adjectives (resulting in about 51,708 vocabularies),
3. Eliminate rare words (those which less than frequency of 64), finally 1,585 words are remain (their frequencies range from 5,150 to 64)

After the extraction on average 32 words are attached to each image in average.

As a result of the operation, 9,681 pairs of images and several words are prepared for the experiment.

3.2 Procedure used in the experiment

This experiment used two-fold cross validation. The data is randomly divided into two groups (4,841 and 4,840). One group is used for learning (i.e., estimating the likelihood of the words). The other is used for recognition (i.e., for the output of words). The same process is repeated after swapping the two groups.

The unit of scale used for vector quantization is defined as the standard deviation of "one-dimensional pulled-down data." The "one-dimensional pulled-down data" is defined as a set of scalar data composed of all the components of feature vectors from the original set of data.

Experiments for reducing the total number of words were also carried out. The procedure used for reducing is as follows:

1. Select (and mark) words randomly by the number of words eliminated.
2. Choose a target word for each marked word through random sampling with replacement from the unmarked words.
3. Identify each selected word with the target word.

3.3 Results

Tables 1 and 2 show the examples of output words (the top 3 words) for images in the recognition group (i.e. 'unknown' images). In Tables 1 and 2, bold words indicate 'hit' words (i.e. originally attached words for the image). Tables 1 and 2 show that words output change depending on images input. However it is difficult with our method to output suitable words in the same manner humans do because of too large a variety of images in this data. In Table 1 and 2, hit words appear more than in the case of random selection (if it is a random selection from a set of words with *uniform frequencies*, the probability is about 3/1585). However the frequencies of words in our data is *not* uniform, the words which have high frequencies tend to appear many times (e.g. 'year', 'Japan').

Table 3 shows the numerical results of the experiment for various scales in vector quantization. In Table 3, the hit rate means the rate of originally attached words in output words. This table shows that scale 4 has the best hit rate. The difference between the hit rate in scale 4 and the hit rate in scale 0 shows that vector quantization is effective. As the scale increases above 4, the hit rate decreases gradually. The scale 22 in Table 3 has only one centroid. In this case, features from images are not considered at all. Therefore, the difference between scale 22 and

[4] Japanese is not divided into words originally.

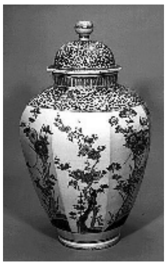
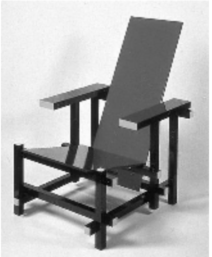
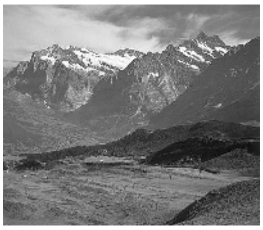
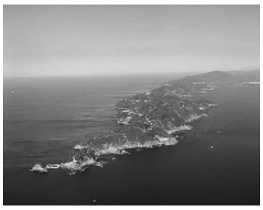
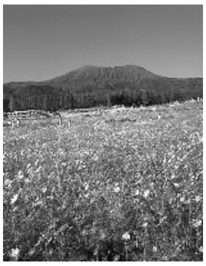
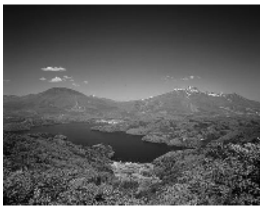
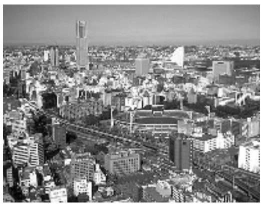
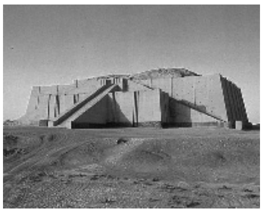
Input image	Output words (top 3)	Input image	Output words (top 3)
	**year**, **Japan**, family		year, **age**, white
	**year**, **many**, family		area, east, shore
	year, **park**, family		**park**, **national**, center
	year, **ten thousand**, **city**		**city**, god, layer

Table 1: Examples of output words for unknown images – part 1. Bold words shows 'hit' words. The image is divided into 3×3, scale = 4.0.

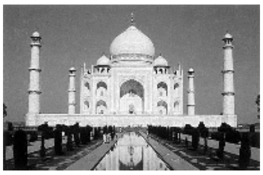
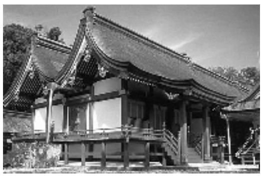
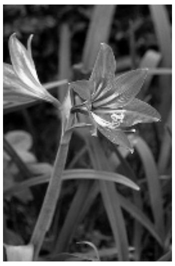
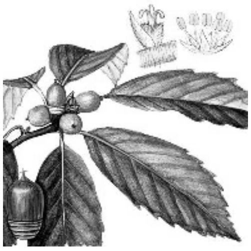
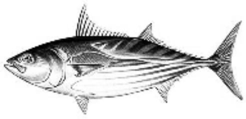
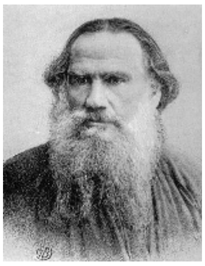
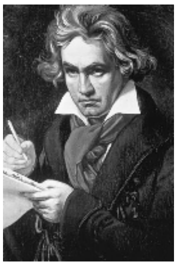
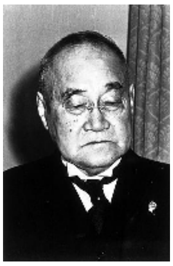
Input image	Output words (top 3)	Input image	Output words (top 3)
	**year**, Japan, China		architecture, **shrine**, represent
	**family**, year, **leaf**		**family**, **leaf**, **flower**
	year, **many**, **Japan**		**year**, **century**, **age**
	**year**, age, **work**		**year**, **Japan**, age

Table 2: Examples of output words for unknown images – part 2. Bold words shows 'hit' words. The image is divided into 3×3, the scale is 4.0.

Scale	Number of centroids		Hit rate (top 3 words)
0	43181	43190	0.31
2	20732	21082	0.35
4	2373	2361	0.40
6	407	404	0.39
12	40	39	0.37
22	1	1	0.33

Table 3: Results for various scales. Two columns in the 'number of centroids' correspond to the two learnings in the two-fold cross validation. The number of divisions is 3 × 3. The hit rate is a mean value for all tests in the two-fold cross validation.

Divisions	Scale	Number of centroids		Hit rate (top 3 words)
1×1	6	164	144	0.37
3×3	4	2373	2361	0.40
5×5	4	3941	4010	0.40
7×7	3	25298	25874	0.41

Table 4: Hit rates for various numbers of divisions. Each scale is the best one for each number of dividing. Two columns in the 'number of centroids' correspond to the two learnings in the two-fold cross validation. The hit rate is a mean value for all tests in the two-fold cross validation

Number of words	Hit rate (top 3 words)	Hit rate (top 10 words)
1585 (original)	0.40	0.25
792	0.37 ± 0.02	0.25 ± 0.01
396	0.37 ± 0.03	0.27 ± 0.01
198	0.39 ± 0.04	0.32 ± 0.02
99	0.48 ± 0.04	0.42 ± 0.02
49	0.62 ± 0.04	0.56 ± 0.02
24	0.77 ± 0.03	0.70 ± 0.02

Table 5: Hit rates for various numbers of words. The number of words is reduced by random identification (an average of 100 iteration). The scale is 4, and the number of divisions is 3×3 for all results.

other smaller scales shows the effect when features from images are considered. Compared to the case for scale 4, there is a 7% advantage.

Table 4 shows the result of the experiment for various number of divisions. In table 4, the more the image is divided, the better the hit rate is. This result cannot verify the optimal number of divisions for the data. However this result at least shows that the hit rate for not dividing (1×1) is inferior to the other.

Table 5 shows the result of reducing the number of words by randomly identifying words with other words. In table 5, the hit rate for the top 10 words increases as the number of words decreases. The hit rate for the top 3 words has almost the same tendency, except for slight decreases in the cases of 792, 396, and 198 words [5]. It can also be seen in table 5 that when the number of words drops below 100, the hit rate reaches about 50% for the top 3 words.

4 Discussions

Our method connects between each feature and word *non-expressly* unlike most existing methods which connect them word by word (e.g. "*leaf* has highly green values ..."). Only *likelihoods* (weights) connect feature vectors within a cluster and words by learning from data. You need not (and should not) care about how they are connected word by word. We think this is the most promising method for annotating words for images in a wide domain because word-by-word descriptions are intractable for a large-scale vocabulary. Of course, not all symbol information can be reduced to features (e.g. historical remarks, dates and times). These types of information are outside the scope of our method.

[5] The reason for this decrease is that the original (1,585) case had a high hit rate because the elimination of low-frequency words (under 64) made it easy to hit correct words.

Experimental results show that the contribution of vector quantization is about 9%. They also show that the effect of considering images, compared with random selection, is about 7%. These percentages cannot be said to be significant or insignificant because they depend on the data used in the experiments. As table 1 shows, a considerably wide domain of images and vocabulary was used in the experiment. These percentages can easily be improved by restricting the domain of images and/or the vocabulary. Table 5 shows the effects of random word reduction. You can estimate that if you want to get a hit rate of about 50% for three-word output, the number of words must be limited to about 100 when this type of data set (from an encyclopedia or similar source with 5,000 images) is used.

However, despite the wide domain of data, our hope is that this method will make real-world data tractable, and that the results produced by this method, when it is combined with a good human-interface system, will help users in mining data.

5 Conclusion

In this paper, we have described a new method for correlating images with key words based on two kinds of processes: dividing images and clustering in feature space. The results from experiments which used data from an encyclopedia confirmed the positive contribution of the two processes.

Future work is needed to develop a method for automatically selecting words depending on given learning data, as well as to determine the optimum size for dividing images depending on the characteristics of image-word databases.

Acknowledgment

The authors would like to thank Hitachi Digital Heibonsha for permission to use the data in *Mypaedia* for this work.

References

Flickner, M., Sawhney, H.S., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D. and Yanker, P. (1995). Query by Image and Video Content: The QBIC System. *IEEE Computer*, 28(9), 23–32.

Kurita, T., Kato, T., Fukuda, I. and Sakakura, A. (1992). Scene retrieval on an image database of full color paintings. *Trans. of Information Processing Society of Japan*, 33(11), 1373–1383 (in Japanese).

Ono, A., Amano, M., Hakaridani, M., Satou, T. and Sakauchi, M. (1996). A flexible content-based image retrieval system with combined scene description keyword. *Proc. IEEE Computer Society, International Conference on Multimedia Computing and Systems '96*, 201–208.

Picard, R.W. and Minka,T.P. (1995). Vision texture for annotation. *Multimedia Systems*, 3(1), 3–14.

Watanabe, Y. and Nagao, M. (1998). Image analysis using natural language information extracted from explanation text. *J. of Japanese Society for Artificial Intelligence*, 13(1), 66–74 (in Japanese).