An Abstract Image Representation Based on Edge Pixel Neighborhood Information (EPNI)

Abdolah Chalechale and Alfred Mertins

University of Wollongong
School of Electrical, Computer and Telecommunications Engineering
Wollongong, NSW 2522, Australia
{ac82, mertins}@uow.edu.au

Abstract. In this paper we introduce a new abstract image representation based on Edge Pixel Neighborhood Information (EPNI). It is applied in image retrieval problem when user query is a fast drawn, rough example. The representation consists of two main elements. A neighborhood vector f and a vicinity table v. The former contains the frequencies of edge pixels with similar directions and the latter holds information about neighboring edge directions. An image similarity measure based on EPNI components is also designed and compared with some other measures known from the literature. Experimental results show a good recognition accuracy in a data set containing a wide range of color images.

1 Introduction

Managing the tremendous number of images and video clips in relevant databases and also on the Web needs more efficient and fast algorithms and tools. Image similarity measurement is one of the most important aspects in a large image database for efficient search and retrieval to find the best answer for a user query. Image and video indexing using a content-based approach plays an important role in finding and accessing minimal information. Recently, this area has attracted many new researches. Representative systems are QBIC [1], Photobook [2], FourEyes [3], MetaSEEk and VisualSEEk [4,5].

In most current content-based image retrieval systems the emphasis is on four clues: color, texture, shape and position of objects. The MPEG7 standard suggests some descriptors for color and texture [6], and for visual shape [7]. Although color and texture are significant features for retrieval purposes, there are some situations where they cannot be used efficiently. For instance, when the query is a rough and quick black and white sketched image and a user is asking the system to find the most similar images to his/her query example, then color and texture lose their original importance. In addition, because the sketched query does not contain a well defined object contour, using shape descriptors may yield undesired results.

In sketch based image and video retrieval situations, when the query is a rude, uncolored example image drawn with some primitive tools, the following methods may generate more acceptable results.

M.H. Shafazand and A M. Tjoa (Eds.): EurAsia-ICT 2002, LNCS 2510, pp. 67–74, 2002.

A) A correlation approach was introduced by Hirata and Kato [8]. In this method, the query and target images are resized to 64*64 pixels, then their edges are extracted by a gradient operator and finally, a global correlation factor (C_t) which is used for similarity measuring is calculated. A modified version of this method is used in QBIC [1].

B) Normalized central moments and skew and rotation invariant functions based on them have been used as powerful tools for shape description (see [9]). Mohamad, Sulong and Ipson [10] used this method for trademark matching in an image retrieval context. They conclude that in scanned b&w images, moment values can be taken as standard features for the matching task. The QBIC system takes advantage of digital moments for shape similarity as well [1].

C) The Hausdorff distance measures the similarity of two sets of points. This distance may be applied to determine the extent to which one image resembles another. Huttenlocher, Klanderman and Rucklidge [11] compared the Hausdorff distance with binary correlation on edge maps and conclude the former works better. They also provide algorithms for computing the Hausdorff distance between all possible relative positions of a binary image and a translated model of the image [12].

D) Histograms of edge directions for representing image information is one of the well known methods in the image retrieval field. Recently, C. S. Won et al. [13] showed that the global and semi-global edge histograms have better retrieval performance than the MPEG-7 recommended local edge histogram descriptor. M. Abdel-Mottaleb [14] used the approach by applying the Canny edge operator to find strong edges in an image and then quantized them into 4 directions. Jain and Vailaya [15] also proposed edge directions as an image attribute for shape description.

In this paper we introduce a new method of feature extraction for retrieval purposes based on edge maps using 1) a vector of neighborhood information of edge pixels and 2) a second-order vicinity table. The vector is used for measuring the similarity between two images and the vicinity table is used for reducing the search space and improving the efficiency of the method. The method is scale independent and yields excellent retrieval results in a data set containing a wide range of images.

In the next section the explanation of the method is provided. Experimental results are presented in Section 3. Conclusions and some directions for future work are finally given in Section 4.

2 Edge Pixel Neighborhood Information(EPNI)

The objective of the proposed approach is to transform the image data into a new structure that supports measuring the similarity between a full colored image and a rough sketch given by a user as a query example in a correct, easy and fast way.

At first, the color image is converted to a gray intensity image by eliminating the hue and saturation while retaining the luminance. Applying the Canny edge operator [16] on this gray scale image results in an edge image I, which is the platform for further feature extraction.

The algorithm uses an edge pixel neighbor diagram (see Fig. 1). In this diagram the center is an edge pixel in I. Considering 8-connectivity, each pixel has, in most cases, up to 4 neighbors as it is an edge point. By numbering the directions as indicated in Fig. 1, each pixel neighborhood is coded with a number n, $0 \leq n \leq 240$, by summing up the direction numbers of its neighbors. For instance, $n = 0$ means a singular point (without any neighbor), a pixel point with two horizontal neighbors has n=17 code and $n = 240$ means a point with 4 neighbors in the directions represented by 128, 64, 32 and 16.

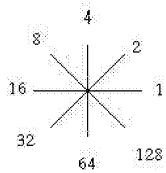

Fig. 1. Edge Pixel Neighbor Diagram.

The frequencies of the neighborhood codes ($n \neq 0$) form a neighborhood vector f with maximally 240 entries. The singular points ($n = 0$) are scarce and not considered. The sum of all edge pixels with the same neighborhood code is stored in the appropriate entry of f. For example, the sum of all edge pixels with two vertical neighbors is stored in entry 68. The numbering scheme simplifies the code finding process since we only set a bit to 1 in the appropriate position in the code byte. The emphasis is on the number of occurrences (frequencies) of each code and not on the code itself. Because for each image, f depends on the size of the image, it is necessary to normalize f to be scale invariant. We found that normalizing f by the size of the image is better than normalizing it by the number of edge pixels as in [15]. Brandt et al. also use this type of normalization in their work [17]. The normalized vectors are defined as:

$$f_i = \frac{\sum_I P_i}{\text{size } I} \tag{1}$$

where $1 \leq i \leq 240$ and P_i is any edge pixel with $n = i$.

The next part of the feature extraction algorithm is a second-order vicinity table v. This table contains information about neighboring edge directions. Most pixel points have only two other neighbors. If P_i is an edge pixel with neighborhood code $n_{1,i}$ and P_j and P_k are its first two neighbors in ascending order with $n_{2,i}$ and $n_{3,i}$ as their neighborhood codes respectively, then there are three neighboring codes for each edge pixel P_i (a triplet). The first code ($n_{1,i}$) is the neighborhood code of the pixel, the second code ($n_{2,i}$) and the third one ($n_{3,i}$) are neighborhood codes of the two neighbors P_j and P_k. See Fig. 2 for an example. If there is only one neighbor point, the third code would be zero. For all

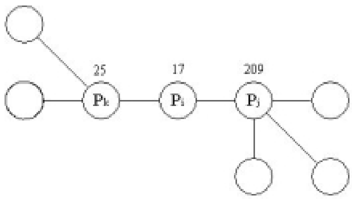

Fig. 2. An example of neighboring edge directions.

triplets $T_i = \{n_{1,i}, n_{2,i}, n_{3,i}\}$ we obtain the frequency of occurrence, denoted as N_i. Sorting with respect to N in descending order results in the vicinity table v whose rows have the structure:

$$v_i = \{n_{1,i}, n_{2,i}, n_{3,i}, N_i\}$$

where
n_1 = neighborhood code,
n_2 = neighborhood code of first neighbor ,
n_3 = neighborhood code of second neighbor and
N = number of triplet $\{n_1, n_2, n_3\}$.

2.1 Similarity Measure

An efficient and effective similarity factor is one essential part in any retrieval approach. Using neighborhood vector f as a feature vector, we can use L_1, L_2 or any other histogram similarity measures to obtain a similarity factor. We designed a new measure to evaluate the similarity between two images. Evaluation method is given in Section 2.3. The goal is to find a similarity factor between a sketched query image q and images in the database (I_d), based on neighborhood vector f. Suppose f_q is the neighborhood vector for query image q and f_d for a database image I_d. To find the similarity between q and I_d we define a measure $\mu(q, I_d)$ as:

$$\mu(q, I_d) = \sum_{i=1}^{240} k_i$$

where

$$k_i = \begin{cases} f_d^i - \theta & \text{if} \quad (f_d^i \geq \theta) \ \& \ (f_q^i < \theta), \\ f_q^i - \theta & \text{if} \quad (f_d^i < \theta) \ \& \ (f_q^i \geq \theta), \\ 0 & \text{else.} \end{cases}$$

θ is a constant value and f^i are the elements of vector f. In Section 3 we will show that this measure is more robust and gives better experimental results than L_1 and L_2.

2.2 Search Space

Comparing q with all images in the database to find $\mu(q, I_d)$, $1 \leq d \leq M$ where M is the number of images in the database, is a time consuming process. Using vicinity tables v and v', representing the images q and I_d respectively, we obtain a correlation factor C as follows:

$$C = \frac{\sum_{r=1}^{j}\sum_{s=1}^{j}(v_{rs} - \bar{v})(v'_{rs} - \bar{v'})}{\sqrt{(\sum_{r=1}^{j}\sum_{s=1}^{j}(v_{rs} - \bar{v})^2))(\sum_{r=1}^{j}\sum_{s=1}^{j}(v'_{rs} - \bar{v'})^2)}}$$

$\bar{v}$ and $\bar{v'}$ are mean values of corresponding vicinity tables. $\bar{v}$ and $\bar{v'}$ are calculated for only j rows and the first 3 columns. j is an arbitrary factor and determines the number of used rows in v. C is in the range of -1 to 1, indicating minimum-to-maximum correlation between the two chosen tables. A lower bound for C limits the number of comparisons. It also improves the overall efficiency as a huge number of non-similar images are disregarded.

2.3 Evaluation Method

The following scoring scheme is defined for similarity measuring evaluation. Let M be the total number of images in the database and $\widehat{q}$ be an image among them which is supposed to be found when providing the query image q. Sorting images by the similarity measure in the descending order will, generally, put $\widehat{q}$ at row k with $0 \leq k \leq M - 1$. $k = 0$ means that the exact image has been found (score 1), small k's mean good and large k's mean poor findings. A score assigning scheme is defined as :

$$S_q = \frac{M - k}{M}$$

For each query q there is an S_q, and for a set of q's there is a set of S_q's. Therefore, the overall efficiency of the similarity measure should be considered upon a set of S_q's as :

$$\eta = \frac{\sum S_q}{\text{number of } q\text{'s}} * 100$$

A large η indicates a good ability to find the most similar answers to the given sketches. This parameter can serve as an evaluation tool for different similarity measures.

3 Experimental Results

To compare the overall efficiencies of different similarity measures, we created a small (but wide range) data set of color JPEG images of 50 images. Different users were asked to sketch rough, black and white queries (16 queries) that resemble images in the data set (see Fig. 3).

For all following measures, we first converted the JPEG images to single band luminance and then applied the Canny operator [16] to gain edge images I_d. For query images also, we determined the edge image q by the same process. For the methods in Table 1, similarity measure were computed as follows:

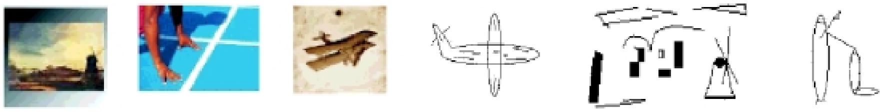

Fig. 3. 3 examples of database images (left) and 3 examples of sketched images (right).

- Correlation. We first resized q and I_d to $64*64$ pixels, then divided them to $8*8$ blocks. Finally the algorithm given in [8] was applied to calculate C_t as the similarity measure.
- Hausdorff. q and I_d were resized to 64*64 pixels, then they were divided into 4 equal sub images and the Hausdorff distance [12] for corresponding sub images (H_1, H_2, H_3, H_4) was obtained. Finally, the minimum was chosen for the similarity measure.
- $L1$ metric (Manhattan-Cityblock). Let f_q and f_{I_d} be the neighborhood vectors of q and I_d. Instead of using all elements of f_q and f_{I_d} we only apply the $L1$ measure to the t most papular edge directions in f_q. For this, f_q is first sorted in descending order to find the t most popular indices. After storing these indices in a set X, the $L1$ similarity measure was calculated as:

  $$L1(q, I_d) = \sum_{i \in X} \left| f_q^i - f_{I_d}^i \right|$$

  We found that the best t in this measure is 13, therefore, X contains only 13 members in our experiments.
- Weighted $L1$. Putting some appropriate weights on the terms of $L1$ summation improves the overall efficiency of the metric. The following set of weights was found to be a good choice for a weighted $L1$ measure ($L1_w$) as the similarity measure of a length-13 vectors f. The weight set puts more emphasis $(12, 8, 6)$ on more important directions in the 13 sorted ones.

  $$w = \{2.5, 12, 8, 6, 4, 3, 2, 3, 4, 6, 6, 8, 2.5\}$$

  The measure then is

  $$L1_w(q, I_d) = \sum_{i \in X} w_i \left| f_q^i - f_{I_d}^i \right|$$

- $L2$ metric (Euclidean). Unlike for the $L1$ metric, all indices in f_q and f_{I_d} have to be considered to maximize the efficiency. Euclidean distance between the two neighborhood vectors as a similarity measure is :

  $$L2(q, I_d) = \sqrt{\sum_{1}^{240} (f_q^i - f_{I_d}^i)^2}$$

- $\mu(q, I_d)$. As explained in Section 2.1, for each q, the μ measure was computed for all I_d's. The parameter θ was set to 0.0089 in finding the k_i's.
- $\mu_r(q, I_d)$. We obtained the measure μ only for those pairs q and I_d that satisfy the constraint $C \geq -0.5$, as explained in Section 2.2, while the number of used rows of vicinity tables (j) was set to 3.

The overall efficiencies η of the considered techniques are presented in Table 1.

Table 1. Efficiencies of different similarity measures.

Technique	Efficiency%
correlation	76.75
Hausdorff	81.126
$L2$	82.75
$L1$	84
$L1_w$	89.75
μ	91.75
μ_r	92.375

It is worthwhile to mention that, in addition to ascending ability of finding the targeted images by techniques in the table, the correlation method is the most time-consuming and μ_r is the fastest one. The next significant point is that when μ_r is used as the similarity measure, only 667 comparisons, instead of 800 (number of queries * number of images) took place. It means 16.625% reduction of search space.

4 Conclusion

We introduced a new algorithm for image similarity measuring in sketch based context based on edge pixel neighborhood information. It is based on two main elements: 1) a feature vector f which includes the frequencies of edge pixels with similar neighborhood directions and 2) a second-order vicinity table v that contains information about neighboring edge directions. We defined also a measure μ for comparing the similarity between two feature vectors. Using table v in addition to f improves the efficiency and also reduces the search space. The paper presented comparative experimental results that showed a great improvement in finding targeted images when using this algorithm and the similarity measure.

The proposed approach is scale invariant and we intent to expand it to be rotation invariant by grouping rotation-similar entries of f . The μ measure could be extended by a weighting concept to further improve the search capabilities.

Acknowledgment. The first author is financially supported by the Ministry of Science, Research and Technology of I.R. Iran.

References

1. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "The QBIC project: querying images by content using color, texture, and shape," in *Proceedings of Spie*, USA, 1993, vol. 1908, pp. 173–187.
2. A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: content-based manipulation of image databases," *International Journal of Computer Vision*, vol. 18, no. 3, pp. 233–254, June 1996.
3. T. P. Minka and R. W. Picard, "Interactive learning with a "society of models"," *Pattern Recognition*, vol. 30, no. 4, pp. 565–581, Apr. 1997.
4. J. R. Smith and S. Chang, "Visualseek: a fully automated content-based image query system," in *Proceedings ACM Multimedia 96.*, NY, USA, 1996, pp. 87–98.
5. M. Beigi, A. B Benitez, and S. Chang, "Metaseek: a content-based meta-search engine for images," in *Proceedings of Spie*, USA, 1997, vol. 3312, pp. 118–128.
6. B. S. Manjunath, J.-R. Ohm, and V. V. Vasudevan, "Color and texture descriptors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 703–715, June 2001.
7. M. Bober, "Mpeg-7 visual shape descriptors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 716–719, June 2001.
8. K. Hirata and T. Kato, "Query by visual example-content based image retrieval," in *Advances in Database Technology - EDBT '92*, Berlin, Germany, 1992, pp. 56–71.
9. A. D. Bimbo, *Visual information retrieval*, Morgan Kaufmann Publishers, 1999.
10. D. Mohamad, G. Sulong, and S. S. Ipson, "Trademark matching using invariant moments," in *Proceedings second Asian Conference on Computer Vision, [ACVV'95].*, Singapore, 1995, vol. 1, pp. 439–444.
11. D. P. Huttenlocher, W. J. Rucklidge, and G. A. Klanderman, "Comparing images using the Hausdorff distance under translation," in *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Los Alamitos, CA, USA, 1992, pp. 654–656.
12. D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, Sept. 1993.
13. C. S. Won, D. K. Park, and S Park, "Efficient use of Mpeg-7 edge histogram descriptor," *Etri Journal*, vol. 24, no. 1, pp. 23–30, Feb. 2002.
14. M. Abdel-Mottaleb, "Image retrieval based on edge representation," in *Proceedings 2000 International Conference on Image Processing*, Piscatway, NJ, USA, 2000, vol. 3, pp. 734–737.
15. A. K. Jain and A. Vailaya, "Image retrieval using color and shape," *Pattern Recognition*, vol. 29, no. 8, pp. 1233–1244, Aug. 1996.
16. J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
17. S. Brandt, J. Laaksonen, and E. Oja, "Statistical shape features in content-based image retrieval," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, Los Almaitos, CA, USA, 2000, vol. 2, pp. 1062–1065.