Hierarchical Indexing Images Using Weighted Low Dimensional Texture Features

Mira Park[1], Jesse S. Jin[1,2], Laurence S. Wilson[3]

1.School of Computer Science and Engineering, University of NSW, mirap@cse.unsw.edu.au

2. School of Information, Technologies University of Sydney, jesse@staff.cs.usyd.edu.au

3. Telecommunications and Industrial Physics, CSIRO, Laurie.Wilson@csiro.au

Abstract

This paper introduces a new method to analyse the image texture and to index the image database. We present a new strategy to reduce the computational time to extract image features with high retrieval accuracy. We also propose a method to reduce the image feature dimension, so any robust indexing methods can be used. By weighting the extracted image features, a system may perceive the image consistent with human perception. We use two spaces to keep the key images and the candidates images for an efficient indexing of the image database.

1. Introduction

Texture is an important component in the perception, classification, identification and segmentation of images for content based image retrieval (CBIR). A variety of techniques have been used for measuring texture similarity. Liu and Picard [22] calculated measures of image texture such as the degree of periodicity, directionality and randomness. Other methods of texture analysis for retrieval include the use of Gabor filters [2, 13] and fractals.

Texture queries can be formulated by selecting examples of desired textures from a palette, or by supplying an example query image. The system then retrieves images from an image database with texture measures most similar in value to the query.

There are three problems to solve: high computational time, handling high dimension data, and comparing images consistent with human perception. The first problem is the high computational time. Since texture has been recognized as an important feature for CBIR, many texture features have been proposed to precisely describe the natural texture properties. Among different texture features, one of the best texture-based feature analysis methods is the multi-resolution Gabor wavelet feature [2]. It can achieve the highest retrieval rate on the entire Brodatz texture database test. However, the drawback of this method is the large computation time in feature analysis. We present a new strategy to compute an image feature with a high retrieval accuracy to reduce the computation time.

The second problem is to handle the high-dimension data. An image database management system needs a multi-dimensional indexing technique since the computed image features have high dimensions. One of the more robust methods, the R*-Tree works well up to 10 dimensions [15], but the overheads for using complex index structures are considerable. One solution to manage the high-dimensional database is to reduce the dimension, so any robust indexing methods can be used. We introduce a method to reduce the image feature dimension using the reward-punishment algorithm.

The third problem is that an ideal CBIR system should compare images in its database with the query in a manner that is consistent with human perception of visual features. We propose means to make the system perceive the image similar with human perception.

In this paper, we present a new image feature extraction with a high retrieval accuracy using Quasi-Gabor filters, introduce a method to reduce the extracted image feature dimension using the reward-punishment algorithm and to weight the extracted image features to make a system perceiving the image consistent with human perception.

The layout of this paper is as follows. In Section 2, we will explain the hierarchical indexing and the dimension reduction. In Section 3, we will describe the image retrieval. In Section 4, we will detail our experiments, and finally, in Section 5, the conclusions of this paper will be presented.

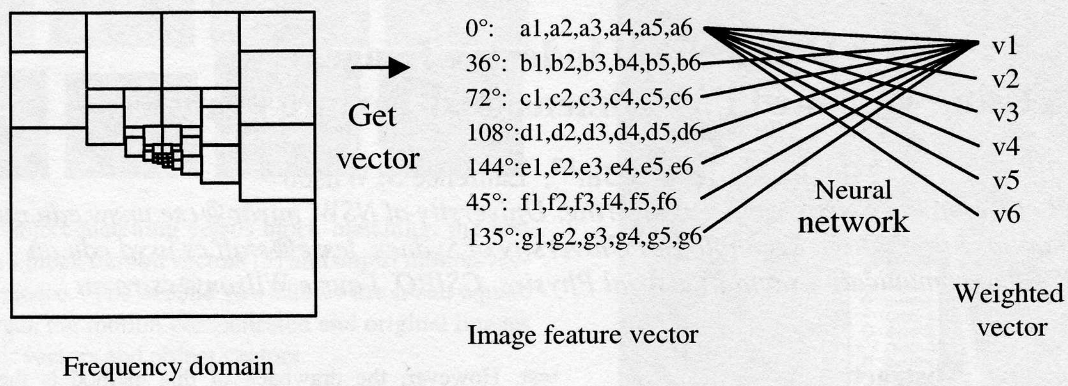


Figure 1. *Extracting the image feature vector using Quasi-Gabor filter and reducing the dimension*

2. Indexing and dimension reduction

2.1. Image database

Our image database consists of image classes. Each class contains two spaces, which are the Key image feature space (KIFS) and the Candidate image feature space (CIFS). KIFS has full dimension image feature vectors selected from CIFS and the weight vector for the class. CIFS includes reduced dimension image feature vectors and they are weighted to increase the discrimination with other images in other classes.

2.2. Quasi-Gabor texture and its extraction

The image classes can be built automatically using some classification algorithms or manually using human perception. Then, the image feature vectors are extracted for each image.

Feature extraction is an important part in the proposed scheme of image retrieval. Image features, such as texture, shape and color, often have high dimension. We will concentrate our discussion on texture features, in particular, multi-channel Quasi-Gabor texture. Feature extractions acquired by this experiment are derived from two methods, which are FFT and Quasi-Gabor filter.

The frequency domain

The DFFT(Discrete Fast Fourier Transform) [7] has been used to convert images to the frequency domain.

Before the two dimensional DFFT is calculated we remove the mean value from the image because the positive peaks of the waveforms are more likely to exceed the maximum level that can be represented. Then we use the Hamming windowing function to reduce DFFT leakage and the side lobes for improving DFFT results. In the Hamming window, the end points don't reach zero. The following equation defines the Hamming window (1):

$$w(n)Hamming = 0.54 - 0.46\cos(2\pi n/N) \qquad (1)$$

In this equation, N represents the number of sine-wave samples, and n equals the sample index:0, 1, 2, … and so on up to N-1. We then perform the two dimensional DFFT for each image.

Quasi-Gabor filters

The texture feature vector used to characterize each image in our experiments is derived with the Quasi-Gabor filter shown in Figure 1. First, the image is filtered through 42 channels by calculation of the energy for each block defined by a combination of one of 6 frequencies (f = 1, 2, 4, 8, 16 and 32) and one of 7 orientations (θ = 0°, 36°, 72°, 108°, 144°, 45° and 135°). The block size lays on 0°, 36°, 72°, 108° and 144° is $f^2 * 2^n / 2^7$ and the block size laid on 45° and 135° is half of the block size laid on other orientations when the size of the image is 2^n x 2^n. A single value, which becomes an entry in a 42-dimension texture feature vector, is then extracted from each block. We take the average value of the magnitude of the filtered image in each block. For using the Quasi-Gabor filter, we do not need any convolution or multiplication of the image with the filter, so it is much faster than using Gabor filters or other filters.

2.3. Dimension reduction and weighting

For the indexing system, the R*-tree can handle 10 dimensions. However a good texture feature is always in high dimension. Therefore it is worthwhile investigating reducing the image feature dimension for efficient indexing and retaining the high retrieval rate. Our system produces a 42 dimensions vector for describing the image texture using Quasi-Gabor filter. The system uses the reward-punishment algorithm (one layer neural network) to select which element best describes the image for the specified class. The training algorithm for the perceptron machine is a simple scheme for the iterative determination of the weight vector w [18]. This scheme, which is frequently called the perceptron algorithm, is stated as follows.

Given two training sets belonging to pattern classes ω_1 and ω_2, respectively, let $\mathbf{w}(1)$ represent the initial weight vector, which may be arbitrarily chosen. Then, at the kth training step:

If $\mathbf{x}(k) \in \omega_1$ and $\mathbf{w}'(k)\mathbf{x}(k) \leq \mathbf{0}$, replace $\mathbf{w}(k)$ by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + c\mathbf{x}(k) \quad (2)$$

where c is a correction increment.

If $\mathbf{x}(k) \in \omega_2$ and $\mathbf{w}'(k)\mathbf{x}(k) \geq 0$, replace $\mathbf{w}(k)$ by

$$\mathbf{w}(k+1) = \mathbf{w}(k) - c\mathbf{x}(k) \quad (3)$$

otherwise, leave $\mathbf{w}(k)$ unchanged, that is

$$\mathbf{w}(k+1) = \mathbf{w}(k) \quad (4)$$

For instance, Class 1 stored a vertical texture and the transformation of the texture showed more important data along the direction 0°. If the system can weight the elements more than other elements it will raise the rate of image retrieval. Therefore the system uses the reward-punishment algorithm to find out the weight for each image feature coefficient. After the system decides all weights for each coefficient, the system ignores all the small weighted coefficients and selects all the large weighted coefficients. As a result the system can reduce the dimension from 42 to 6 (see Figure 1). The weight vectors are different for each class. If two images, which are not recognized as similar images by a computer, want to be a same class, the system weights the images with the class weight vector. The system then perceives the images as the same class.

2.4. KIFS and CIFS

The reduced and weighted image feature vectors in a class are stored in the CIFS. The vectors in CIFS of each class can be classified into subclasses using simple distance calculation algorithms. KIFS is formed with several key vectors randomly selected from each subclass in a class.

3. Image retrieval

Nearest neighbour algorithms have been widely used to find the nearest neighbours of an unidentified query image within a hyper-sphere of predefined radius in order to determine its true class. *k_nn* (k nearest neighbour) rule has been described by Singh [23] as follows:

- Out of N training vectors, identify the k nearest neighbours, irrespective of class label. k is chosen to be odd.
- Out of these k samples, identify the number of vectors, k_i, that belong to class ω_i, i=1, 2, …M. Obviously $\Sigma k_i = k$.
- Assign x to the class ω_i with the maximum number k_i of samples.

As mentioned in Singh [23] a conflict can occur. A conflict means that equal number of training neighbours are found for more than one class in the class determination of the query image. Conflicts can be resolved by increasing the size of the hyper-sphere.

Our system uses *k_nn* to find a true class from KIFS for a query image and applies the weight of the class on the query image feature vector. Then the reduced 6 dimensions query image feature vector is compared with the candidate image feature vectors in CIFS of the class using Euclidean distance and the most closest images to the query image are retrieved.

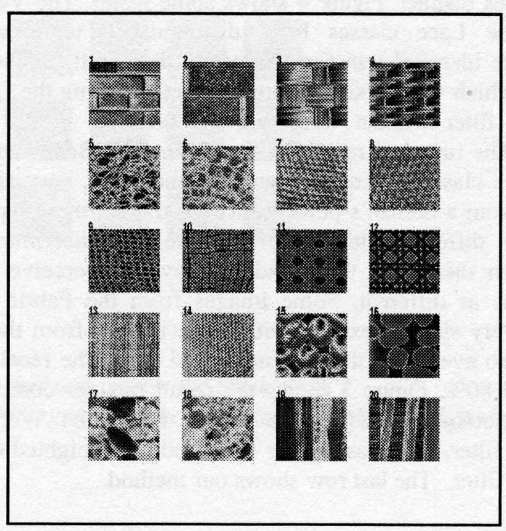


Figure 2. *Images in texture database*

	Brick %	Basket %	Water %	Reptile_skin %	Net %	Lace %	Fabric %	Beans %	Pebble %	Vertical %
Gabor(G)	72	34	68	50	62	92	52	38	44	44
Weighted G	82	92	96	60	70	96	100	96	64	88
Quasi Gabor(QG)	58	54	30	46	60	98	48	26	32	92
Weighted QG	100	100	98	98	82	100	78	100	96	100

Figure 3. *The recall rate of the image retrieval for each class*

4. Performance analysis

Our texture database consists of 10 classes and each class contains two similar 640*640 images. Each image is divided into 25 128*128 nonoverlaping subimages. Therefore there are 10 KIFS in our system and each of them contains 20 key image feature vectors. We formed only one CIFS, which contains all 500 image feature vectors for our experiment. Therefore for most query images there are 50 'matches' available out of a database of 500 images. Figure 2 shows 10 kinds of images stored in our texture database. Image 1 and 2 are the same class to represent the Brick class, and images 9 and 10 for the Net class. These two classes have clear vertical and horizontal lines and this effect the frequency domain, so the system often retrieves both of them as similar images. However, while the classes are training, they have different weights for their image feature coefficient and the difference between the classes becomes distinct. Figure 4 shows some result. The Vertical and the Lace classes have distinguished textures and produce identical transformations, so the recall for the raw data, which was obtained from the method using the Quasi-Gabor filter without weighting the features, is still very high. The raw data recall for the Water, the Beans and the Pebbles classes are quite low since they look very similar even from a human's perception. We trained these patterns in the different classes to increase the discrimination between them. We then made the system perceive these patterns as different. Some images from the Fabric class have very similar textures with some images from the Net class, so even after the system trained them, the recall was around 80%. Figure 3 shows the recall rate for comparing four methods including the Gabor filter, the Weighted Gabor filter, the Quasi Gabor filter and the Weighted Quasi Gabor filter. The last row shows our method.

5. Conclusion

In this paper, we achieved a new image feature extraction using Quasi-Gabor filter. The system of the reduction of the feature dimension is a robust with very low computation time even without any filter convolution, so the system is suitable for a real time system. The system also has a low dimension feature, making it more secure whatever indexing method is used. For efficiency, the proposed image retrieval method can adopt a spatial searching structure, such as R-Trees, to accelerate the retrieval of similar images for a query image.

Future research will include more features in a vector to describe the image texture clearly. It will then be trained to select only those elements that are more important than others to reduce the dimension. Gabor filter, wavelet transformation or other methods can be used to produce image features if the computation time is not important. Our system may also be able to train the features to reduce the dimension using a multi-layer neural network to improve the result. This paper has presented the possibility of a fast image retrieval system with low dimension. This method could be expanded to other areas such as natural images or medical images.

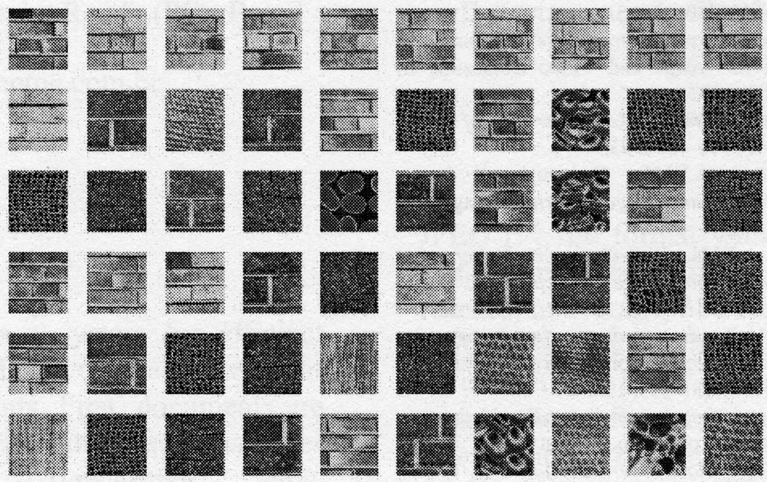
References

[1] Kato, T., *Database architecture for content-based image retrieval* in image storage and Retrieval System(jambardino, A A and Niblack, W R, eds), Proc SPIE 1662:112-123, 1992.

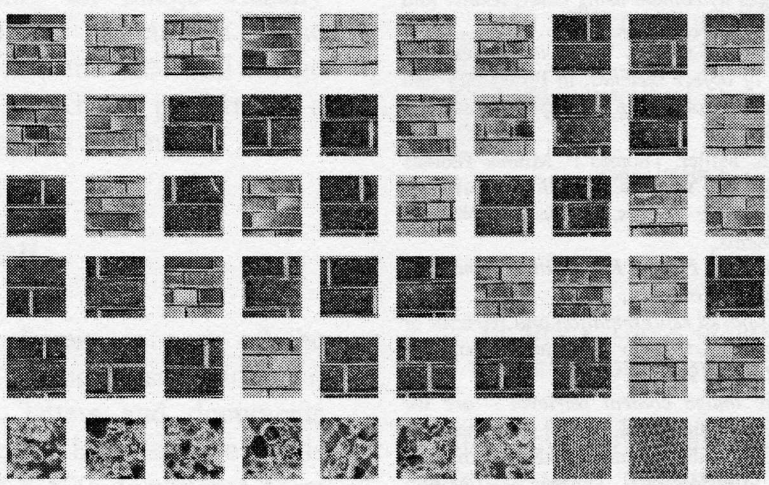
[2] Manjuanth, B.S. and Ma, W.Y., *Texture features for browsing and retrieval of image data*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(8):837-842, 1996.

[3] R.W. Picard, T. Kabir, and F.Liu, *Real-time recognition with the entire Brodatz texture database*, Proc. IEEE Conf, CVPR, 638-639, New York, June 1993.

[4] Gudivada, V.N., Raghavan, V.V., *Content-based image retrieval systems.* Guest editor's introduction. IEEE computer, 28(9):3-7, 1995.



(a)



(b)

Figure 4. *The query image is in the top lefthand corner and is from the Brick class.*
(a) Retrieval of 60 images using raw data. (b) Retrieval of 60 images using the reduction of the images feature dimension

[5] Guttman, A., *R-trees: A dynamic index structure for sparial searching*. In: Proceedings of ACM-SIGMOD, 47-57, 1984.

[6] Lin, K.I., Jagadish, H.V., Faloutsos, C., *The TV-tree: An index structure for high-dimensional data*. VLDB Journal, 3(4):517-542, 1994.

[7] Cooley, J. W. and Tukey, J. W., *An algorithm for the machine calculation of complex Fourier series*, Mathematics of Computation, 19(90):297-301. 1965.

[8] Swan.M.J., *Interactive indexing into image database.* In Storage and Retrieval for Image and Video Databases, In Proc. SPIE, 1908:95-103, 1993.

[9] Chang, S.f., *Content-based indexing and retrieval of visual information. SPMag,* 14(4):45-48, 1993.

[10] Gevers, T., Smeulders, A.W.N., *Content-based image retrieval by viewpoint-invariant image indexing.* Image and Vision Comput, 17(7), 1999.

[11] Gevers,T., Smeulders, A.W.N., *PicToSeek: combining color and shape invariant features for image retrieval.* IEEE Trans. Image Process., 2000.

[12] Swets D & Weng J. *Using discariminant eigenfeatures for image retrieval.* IEEE transactions on pattern analysis and Machine Intelligence, 18(8):831-836, 1996.

[13] Jesse S. Jin, *Depth Acquisition in two dimensional computer vision*, PhD thesis, University of Otago, August 1992.

[14] Daugman J. *Uncertainty relation for resolution in space, spatial-frequency, and orientation optimaized by two-dimensional visual cortical filters.* Journal of the Optical Society of America, 2(7):1160-1169, 1990.

[15] Niblack W, Barber R, Equitz W, Flickner M, Glasman E, Pertovic D, Yanker P, Faloutsos E & Taibin G. *The QBIC Project: Querying images by content using color, texuter and shape.* proceedings of the SPIE, 1908:173-187, 1993.

[16] A. Del Bimbo, P.Pala, and S.santini, *Visual image retrieval by elastic deformation of object sketches*, in Proc IS&T and SPIE Storage and Retrieval of Image and Video Databases VI, San Jose, CA, Jan. 1998

[17] Ingemar J. Cos, Matt L. Miller, Thomas P. Minda, Thomas V.Papathomas and Peter N, Yianilos, *The Bayesian Image Retrieval System, PicHunter: Theory, Implimentation, and Psychophysical Experiments*

[18] J. I. Tou, and R. C. Gonzalez, *Pattern Recognition Principles,* Addison-Wesley Publishing Company, 1974.

[19] Aigrain, P et al *Content-based representation and retrieval of visual media - a state-of-the-art review* Multimedia Tools and Applications, 13(3):197-202, 1996.

[20] Eakins. J. P *Automatic image content retrieval – are we getting anywhere?* Proceedings of Third International Conference on Electronic Library and Visual Information Research(ELVIRA3), DeMontfort University, Milton Keynes, 123-135, 1996.

[21] Idris F and Panchanathan, S *Review of image and video indexing techniques* Journal of Visual Communication and Image Representation, 8(2):146-166, 1997.

[22] Liu F & Picard R. *Periodicity, directionality and rendommness: Wold features for iamge modelling and retrieval.* IEEE Transaction on Pattern anaysis and machine Intelligence, 18(7): 722-733, 1996.

[23] S. Singh, J.F. Haddon, M. Markou, Nearest neighbour strategies for image understading, Proc. Workshop on Advanced Concepts for Intelligent Vision, Systems, Baden-Baden, 2-7, August, 1999.