한국ITS학회논문지
제8권, 제1호 (2009년 2월)
pp. 127∼135

Interest Point Detection Using Hough Transform and Invariant Patch Feature for Image Retrieval

Ahmad Nishat* 안 영 은** 박 종 안***
(Ahmad Nishat) (Young-Eun An) (Jong-An Park)

Abstract

This paper presents a new technique for corner shape based object retrieval from a database. The proposed feature matrix consists of values obtained through a neighborhood operation of detected corners. This results in a significant small size feature matrix compared to the algorithms using color features and thus is computationally very efficient. The corners have been extracted by finding the intersections of the detected lines found using Hough transform. As the affine transformations preserve the co-linearity of points on a line and their intersection properties, the resulting corner features for image retrieval are robust to affine transformations. Furthermore, the corner features are invariant to noise. It is considered that the proposed algorithm will produce good results in combination with other algorithms in a way of incremental verification for similarity.

Key words: Image indexing, multimedia/Image databases, image retrieval, multimedia search, content based image retrieval, image content retrieval

Ⅰ. Introduction

Research in the area of ‘Content based image retrieval’ (normally abbreviated as CBIR) and CBIR systems has come a long way since its first use by T. Kato [1] to describe experiments into automatic retrieval of images from a database, based on colors and shapes. Some very interesting observations were made by Horst Eidenberger [2] in 2003 that Visual information retrieval (VIR) is a research area with more than 300 scientific publications every year. In the year 2003 there were more than thousand papers in this area of research. In 2002 the IEEE alone published more than 700 retrieval papers. Since 1994 more than hundred papers have been published every year.

The best review of CBIR till 2000 is provided by Arnold et. al. [3]. They reviewed 200 references in content based image retrieval. They discussed the working conditions of content-based retrieval : patterns of use, types of pictures, the role of semantics, and the sensory gap. They reviewed algorithms for retrieval sorted by color, texture, and local geometry. Similarity of pictures and objects in pictures is reviewed for each of the feature types, inclose connection to the types and

* 주저자 : 조선대학교 정보통신공학과 박사과정
** 공저자 : 조선대학교 정보통신공학과 박사과정
*** 교신저자 : 조선대학교 정보통신공학부 교수
† 논문접수일 : 2008년 10월 21일
† 논문심사일 : 2008년 11월 24일
† 게재확정일 : 2008년 11월 29일

means of feedback the user of the systems is capable of giving by interaction. They also presented their view on : the driving force of the field, the heritage from computer vision, the influence on computer vision, the role of similarity and of interaction, the need for databases, the problem of evaluation, and the role of the semantic gap.

Markus et. al. [4] presented a comparison of different techniques for consecutive stages of a CBIR system. Veltkamp and Tanase [5] reviewed 43 content-based image retrieval systems. They evaluated how the user can formulate a query, whether and how relevance feedback is possible, what kind of features are used, how features from query image and data base image are matched, what indexing data structures are used, and how the retrieval results are presented to the user. Deb and Zhang [6] presented an up-to-date review of various content-based image retrieval techniques.

Geometric shapes and corners form a major paradigm in the evaluations and identification of graphical information by brain (human perception) [7]. The proposed algorithm is a new idea with a perspective that it can be used in combination with other available techniques for multiple feature selection due to the prominence of corner shapes in visual information processing by brain. The algorithm uses feature set based on corner shapes detected through finding line intersections for object retrieval.

The paper sequence is organized conventionally, discussing prior work, proposed idea and its various aspects with a discussion on the experimental results with concluding remarks in the end followed by relevant references consulted while preparing this paper.

Ⅱ. Prior Work

There is an abundance of literature on corner detection. Moravec [8] observed that the difference between the adjacent pixels of an edge or a uniform part of the image is small, but at the corner, the difference is significantly high in all directions. Harris [9] implemented a technique referred to as the Plessey algorithm. The technique was an improvement of the Moravec algorithm. Beaudet [10] proposed a determinant (DET) operator which has significant values only near corners. Kitchen and Rosenfeld [11] presented a few corner detection methods. The work included methods based on gradient magnitude and gradient direction, change of direction along edge, angle between most similar neighbors, and turning of the fitted surface. Lai and Wu [12] considered edge-corner detection for defective images. Tsai [13] proposed a method for boundary-based corner detection using neural networks. Ji and Haralick [14] presented a technique for corner detection with covariance propagation. Lee and Bien [15] applied fuzzy logic to corner detection. Mokhtarian [16] used the curvature-scale-space (CSS) [17] technique to search the corner points. The CSS technique is adopted by MPEG-7.

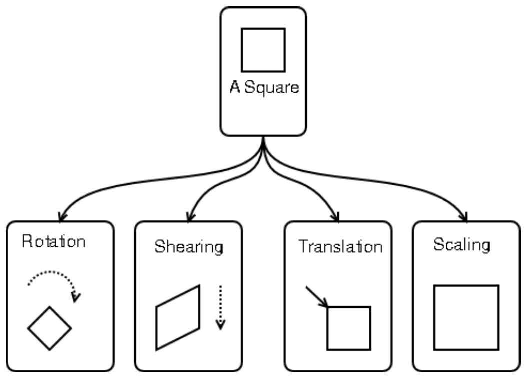
The Hough transform later introduced in generalized form for lines and curve detection has been focus of research interest after it was popularized by the journal article by D.H. Ballard Davies applied the generalized Hough transform to corner detection. Diou, A. et al. proposed an analytical approach for the calculation of the theoretical Hough transform on standard images for research of straight lines. Anastasios & Nikos proposed the Inverse Hough Transform. Fei Shen & Han Wang used modified Hough transform for corner detection. Yu-Hua Gu presented corner based feature extraction for object retrieval using smoothed object boundary curve and 2D rotationally symmetric band pass filter for detecting sharp angles (corners) and used the corner information for object matching and retrieval. For object matching they used normalized arc-lengths between adjacent corners, corner to centroid distances and object boundary curves modeled by a constrained active B-spline curve model.

Ⅲ. Proposed Idea/Algorithm

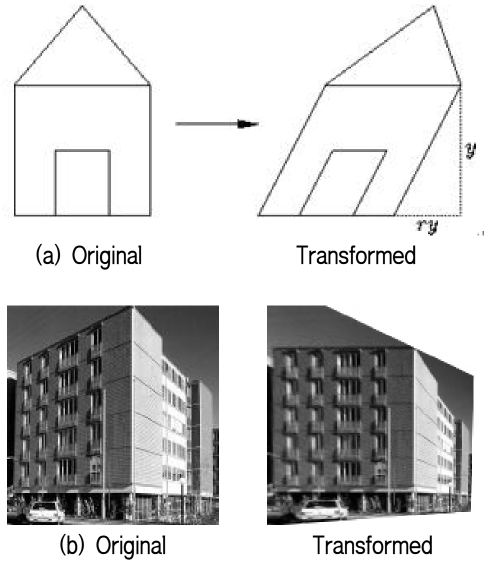
For the purpose of our algorithm, we use the corner feature characteristics in an image, obtained through a neighborhood operation for identifying an image and its subsequent indexing for image retrieval. We define the corner as an intersection of two or more straight lines. So in order to find corners first we need to find straight lines in an image. The reason for using the line intersections for defining corners is that lines are invariant to various affine transformations.

An Affine transformation is a geometrical transformation which is known to preserve the parallelism of lines but not lengths and angles. We can also say that these preserve co-linearity of points and their intersection properties. In other words, three points that lie on a line will continue to lie on a line after application of an affine transformation, as shown in figure 2 (a & b). Affine mappings are of the form Ax+bw here A is an [n x n] square matrix and x and b are vectors in R^n.

The basic affine transformations rotation, shearing, translation and scaling; are shown in figure 1. More general type of affine transformations shown in figure 1 can also be applied in combination as well as selective. For example scaling in only one axis can be termed as squeezing and we can combine rotation for example with this. In this way these combined transformations



<Fig. 1> Common Affine transformations



<Fig. 2> Change of line lengths & Angles after shear transformation

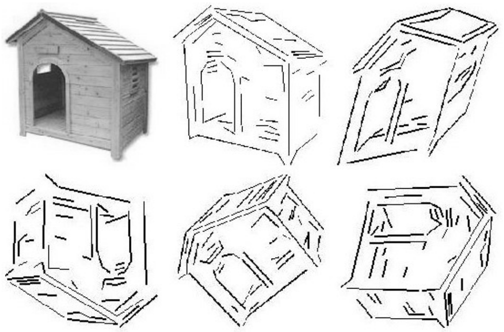
are responsible for completely changing the scene orientation. Camera angle plays a vital role in multiple views of the same image and in order to find visually similar images, it is very important to overcome this aspect by using features which are invariant to affine transformations.

Figure 2(a) below shows a sketch and its shape after applying a sheer transform. After the transform the angles and line lengths have changed but the lines and their intersection relationships remain the same. In figure 2(b) two images of the same building are shown while second one has been stretched or sheer transformed. The preservation of lines and their intersections is obvious.

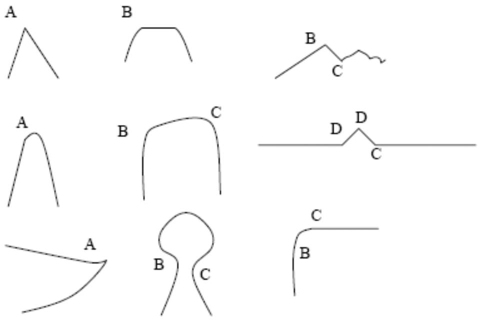
1. Line Detection

Hough transform can be efficiently used to search the straight lines in the images using the parameterized line equation (1).

$$\rho = x\cos\theta + y\sin\theta \qquad (1)$$



<Fig. 3> Detected lines in transformed image



<Fig. 4> Four types of corners

Each line in the image can be associated with a couple (ρ, θ) which is unique if $\theta \in [0, \pi]$ and $r \in R$, or if $\theta \in [0, 2\pi]$ and $r \geq 0$. The (ρ, θ) plane is sometimes referred to as Hough space. From the Hough space the lines can be found using the inverse Hough transform.

The figure 3 below shows an original image and lines detected using the Hough transform in the original and affine transformed images of the original. The co-linearity of points has been preserved in the affine transformed image.

2. Corner detection

Ideally, a corner is an intersection of two straight lines. However, in practice, corners in the real world are frequently deformed with ambiguous shapes. As corner represent certain local graphic features at abstract level, corners can intuitively be described by some semantic patterns (see Fig. 4). A corner can be characterized as one of the following four types:

- Type A: A perfect corner as modeled in [18], i.e., a sharp turn of curve with smooth parts on both sides.
- Type B: The first of two connected corners similar to the END or STAIR models in [18], i.e., a mark of change from a smooth part to a curved part.
- Type C: The second of two connected corners, i.e., a mark of change from a curved part to a smooth part.
- Type D: A deformed model of type A, such as a round corner or a corner with arms neither long nor smooth. The final interpretation of the point may depend on the high level global interpretation of the shape.

Figure 4 shows some examples of the four types of the corner. It is obvious from the Figure that the corner points at very small level are the intersection points of the two straight lines.

Because of many intersections of lines, false corners are also detected. To avoid false candidates, the detected corners whose vicinity does not contain any edge point are discarded.

3. Feature Extraction

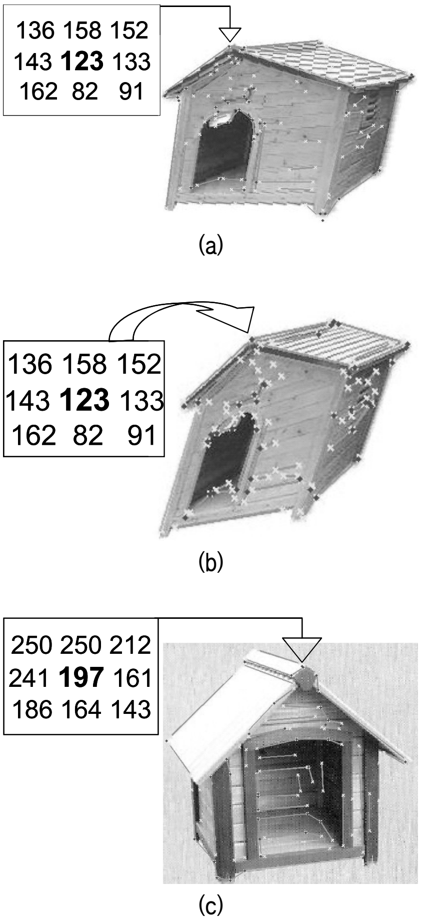
From the corner point information, feature vector is extracted using a neighborhood operation, for image retrieval. The neighborhood operation can be represented by the following pseudo code:

```
Visit each point p (corner) in the image data and do
    {
    N = a neighborhood or region of the image data
    around the point p
    result(p) = f(N)
    }
```

N has been tested with (3x3 & 5x5), with no significant difference on a medium sized database. f (N) arranges the neighborhood pixels in an ascending order to cater for any positional variations in the neighborhood pixels due to some affine transform.

So each image results into a row matrix of 9 columns each with a range from 0~255 in case of a gray level and a triplet in case of taking RGB. Denoting detected corner by 'C' and neighborhood pixel by 'p' we can write:

$$C = p_1, p_2, p_3, \cdots\cdots\cdots\cdots. \cdots\cdots\cdots\cdots. p_9 \quad \text{Where } p = 0 \sim 255 \tag{2}$$



<Fig. 5> Corner neighborhood pixel values

Figure 5 (a) and (b) above shows the corner neighbor hood values of the same image which has been transformed by horizontal scaling and (c) for a different image. In the above figure it can be seen that the corner characteristics remain the same even if the image is transformed. In other words, we will get the same neighborhood values in case of a transformed image.

Considering one corner only, this means that there are a total of $(256)^9$ combinations available to discriminate each corner point on one to one correspondence basis. For each image 'I', we obtain a corner feature matrix(CFM) which is (no.ofcornersx9) in size. Mathematically it can be written as

$$CFM \equiv C_{i=1}^{n}\{C_i \in I \mid C_i = p_{j=1}^{m}\} \tag{3}$$

Where m = 9 for 3x3 matrix. In this case the discrimination power of the corner feature matrix can be given by:

$$[\text{Pixel value}]^{(\text{noofcornersx9})} \tag{4}$$

This feature matrix works well with rotational and scaling transformations, as the ordering of line segment can be based on the length of line segments. But as the length of line segments can change due to affine transformations, there is no reference to keep the ordering of the corner points. In order to cater for this we tried to use the summation of the corner neighborhood values. The resulting feature matrix changes as following:

$$CFM \equiv C_{i=1}^{n}\{C_i \in I \mid C_i = \sum_{j=1}^{m} p\} \tag{5}$$

So each corner will have a value from 0~2304. This resulting feature matrix is a row matrix with number of columns corresponding to the number of corners i. e., [1 x no of corners]. In order to overcome the corner ordering problem we arranged the matrix in descending order. So the final feature matrix has following characteristics:

- It is a row matrix with [1 x no of corners] size.
- Each column has a value from 0~2306, with $(2306)^{(\text{no of corners})}$ as discrimination power.

Values of only four corners from images whose one corner has been shown in figure 5 (b) and (c) are given in below tables. The sum of neighborhood value forms the feature matrix used for comparing similarity measure between images.

Corner	Corner Neighborhood pixels									Sum
1.	136	158	152	143	123	133	162	82	92	1181
2.	250	251	253	230	232	242	147	157	175	1937
3.	224	248	252	227	237	231	141	126	133	1819
4.	226	233	240	150	207	214	93	111	146	1620

Corner	Corner Neighborhood pixels									Sum
1.	250	250	212	241	197	161	186	164	143	1804
2.	244	216	184	193	198	179	99	143	160	1616
3.	179	178	194	177	166	151	177	176	139	1537
4.	19	48	184	15	61	208	11	72	226	844

Ⅳ. Experimental Results

Instead of exact matching, content-based image retrieval calculates visual similarities between a query image and images in a database. Accordingly, the retrieval result is not a single image but a list of images ranked by their similarities with the query image. Many similarity measures have been developed for image retrieval based on empirical estimates of the distribution of features in recent years. Different similarity/distance measures will affect retrieval performances of an image retrieval system significantly.

Let D be the image database and Q be the query image. The distance measure between D and Q, has been taken from [16]. We obtain a permutation of the images in D based on Q i.e., assign rank (I) $rank(I) \in |D|$ for each $I \in D$, using some notion of similarity to Q. This problem is usually solved by sorting the images $Q' \in D$ according to $|f(Q') - f(Q)|$, where $f(.)$ is a function computing feature vectors of images and $|.|$ is some distance measure defined on feature vectors. L1 and L2 distance measures are commonly used when comparing two feature vectors. Here L1 distance measure is used for comparing the proposed algorithm because it is simple and robust. The formula used for comparing the two feature vectors is given as:

$$|I - I'| = \frac{\sum |A_i - B_i|}{\sum |1 + A_i + B_i|} \tag{2}$$

The algorithm was tested on a medium sized database of 3000 pictures and the results are promising. We will now discuss the results obtained using the proposed algorithm using the above described similarity measure.

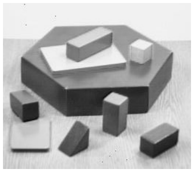
Fig. 6 and Fig. 7 show the query image and the results obtained. The feature matrix derived in this algorithm for an image can be simply described as a corner characteristics signature also giving the number of similar corner geometries. The idea was based on the assumption that similar images will have similar corner signatures.

The database not only consisted of a wide variety of images but also affine transformed images of each normal image. As discussed before in the first paragraph of section 3.1, that each cell in the 3x3 matrix chosen for representing the corner geometry can present a line within 45 degree. During the query, the images which were stretched less than 45 degrees were considered similar. First three results in Fig. 6 and first five results in Fig. 7 are those images which were slightly stretched. The query image in Fig. 6 (a) is also slightly transformed.

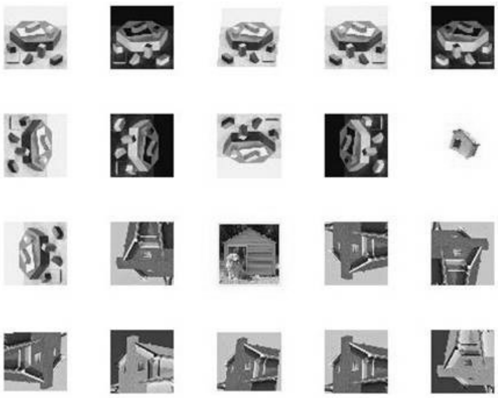
(a) Query Image

(b) Query results

<Fig. 6> Image retrieval test1 results



(a) Query Image



(b) Query results

<Fig. 7> Image retrieval test2 results

In order to check for rotation a matching algorithm was used to check for 100% matching by circular shifting the histogram values. As a result image 4 in the Fig. 6 (b) was identified as similar, where as the image 15 in the same result which is inverted and rotated found a far place in similarity measurement.

The size of the feature vector is small and thus the query time is very less. However, the algorithm gives better results in cases where it can extract a good number of corners. In many cases of pictures depicting nature, such as blue sky with birds or a lake scene with birds, does not give good results.

Ⅴ. Performance Evaluation

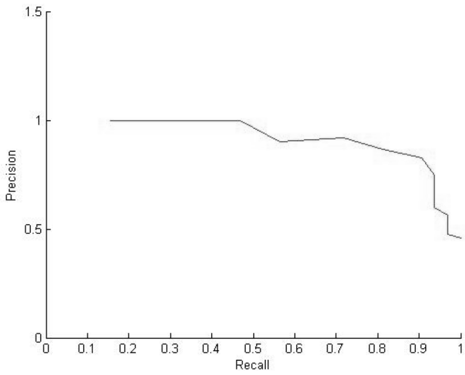
Two traditional measures for retrieval performance in the information retrieval literature are precision and recall. Precision is defined as the percentage of retrieved images that are actually relevant

$$Precision = \frac{Retrieved\ \&\ Relevent}{Total\ \#\ of\ retrieved} \tag{3}$$

Recall is defined as the percentage of relevant images that are retrieved

$$Recall = \frac{Retrieved\ \&\ Relevent}{Total\ \#\ of\ retrieved} \tag{4}$$

Given a query, high precision implies that very little irrelevant images have been retrieved and high recall implies that much of what is relevant in the database have been retrieved. Lack of precision can be compared to a type 2 error (false alarm) and deficiency in recall for a given search is comparable to type 1 error (misdetection). For performance evaluation, one can plot precision and recall as a function of the number of images retrieved as well as the precision versus recall curves for different numbers of images retrieved. To evaluate the overall retrieval performance (precision and recall), first, the database is queried with each of the



<Fig. 8> Performance evaluation

images in test database consisting of images from different visual classes, then average precision and recall percentages are computed for the entire database. To rank-order the database images, distance measures discussed above are used. Fig. 8 shows the averaged precision and recall for the entire database.

Ⅵ. Conclusion

A new technique for image retrieval based on corner shapes in an image has been presented. The proposed feature set is significantly smaller in size compared to the algorithms using color features and thus is computationally very efficient. The corners have been extracted by finding the intersections of the detected lines. As the affine transformations preserve the co-linearity of points of a line and their intersection properties, the resulting corner features for image retrieval are robust to affine transformations. As the corners may not be a very dominant aspect in every image, it is considered that the proposed algorithm will produce best results in combination with other algorithms in a way of incremental verification for similarity.

References

[1] T. Gevers and A. W. M. Smeulders, “PicToSeek: Combining color and shape invariant features for image retrieval,” *IEEE Trans. Image Processing*, vol. 9, no. 1, pp. 102-119, Jan. 2001.

[2] L. Li and W. Chen, “Corner detection and interpolation on planar curves using fuzzy reasoning,” *IEEE Trans. Pattern Analysis and Machine Vision*, vol. 21, no. 11, pp. 1204-1210, Nov. 1999.

[3] S. Panchannathan and C. Huang, “Indexing and retrieval of color images using vector quantization,” *SPIE Proc. Applications of Digital Image Processing*, vol. 3808, pp. 558-568, July 1999.

[4] T. Uchiyama, M. Yamaguchi, N. Ohyama, N. Mukawa, and H. Kaneko, “Multispectral image retrieval using vector quantization,” *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 30-33, Oct. 2001.

[5] F. Idris and S. Panchanathan, “Storage and retrieval of compressed images,” *IEEE Consumer Electronics*, vol. 41, no. 3, pp. 937-941, June 1995.

[6] D. G. Sim, H. K. Kim, and R. H. Park, “Fast texture description and retrieval of DCT based compressed images,” *Electron. Lett.*, vol. 37, no. 1, pp. 18-19, Jan. 2001.

[7] J. Z. Wang, G. Wiederhold, and O. Firschain, “Wavelet-based image indexing techniques with partial sketch retrieval capability,” *Proc. the Forum on Research and Technology Advances in Digital Libraries*, pp. 13-24, May 1997.

[8] R. Rickman and J. Stonham, “Content based image retrieval using colour tuple histograms,” *Proc. SPIE Storage and Retrieval for Image and Video Databases*, vol. 2670, pp. 2-7, July 1996.

[9] P. V. C. Hough, *Method and means for recognising complex patterns*, U.S. Patent no. 3069654, 1962.

[10] R. O. Duda and P. E. Hart, “Use of the Hough transform to detect lines and curves in pictures,” *Commun. ACM*, vol. 15, pp. ll-15, Sept. 1972.

[11] D. H. Ballard, “Generalizing the Hough transform to detect arbitrary shapes,” *Pattern Recognition*, vol. 13, pp. 111-122, July 1987.

[12] E. R. Davies, "Application of the generalised Hough transform to corner detection," *IEE Proc. Computers and Digital Techniques*, vol. 135, no. 1, pp. 49-54 Jan. 1988.

[13] D. A, V. Y, Santo, and C, "The Hough transform-a new approach," *Proc. IEEE Conf. Industrial Electronics, Control, and Instrumentation*, vol. 3, pp. 1612-1617, Aug. 1996.

[14] A. L, Kesidis and N. Papamarkos, "On the inverse Hough transform," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol 21, no. 12, pp. 1329-1343, Dec. 1999.

[15] F. Shen and H. Wang, "Corner detection based on modified Hough transform," *Elsevier Science Inc*, vol. 23, no. 8, pp. 3-21, June 2002.

[16] A. Rattarrangsi and R. T. Chin, "Scale-based detection of corners of planar curves," *IEEE Trans. Pattern Analysis and Machine Vision*, vol. 14, no. 4, pp. 430-449, Apr. 1992.

[17] J. Huang, S. R. Kumar, M. Mitra, W. J. Zhu, and R. Zabi, "Image indexing using color correlograms," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1024-1036, June 1997.

[18] Y. H. Gu and T. Tjahjadi, "Corner based feature extraction for object retrieval," *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 119-123, Oct. 1999.

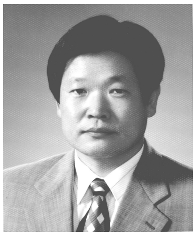
저자소개

Nishat, Ahmad

1992년 : Pakistan Navy Engineering College 공학사
2006년 : BAHRIA UNIVERSITY 공학석사
2008년 : 조선대학교 정보통신공학과 박사수료

안 영 은 (An, Young-Eun)

2004년 : 조선대학교 자연과학대학 수학.전산통계학과(전산통계전공) 공학사
2006년 : 조선대학교 정보통신공학과 공학석사
2008년 : 조선대학교 정보통신공학과 박사수료



박 종 안 (Park, Jong-An)

1975년 ~ 현재 : 조선대학교 정보통신공학부 교수
1990년 ~ 1991년 : 영국 Surrey 주립대학 전기 & 전자공학과 객원교수
1983년 ~ 1984년 : 미국 Massachus sette주립대학 전기&전자공학과 객원교수
1975년 : 조선대학교 전자공학과 공학사
1978년 : 조선대학교 전기공학과 공학석사
1986년 : 조선대학교 전기공학과 공학박사