한국ITS학회논문지
제8권, 제3호 (2009년 6월)
pp. 122~130

Image Feature Representation Using Code Vectors for Retrieval

Ahmad Nishat* 조 혜** 박 종 안*** 박 승 진**** 양 원 일*****
(Ahmad Nishat) (Hui Zhao) (Jong-An Park) (Seung-Jin Park) (Won-Il Yang)

Abstract

The paper presents an algorithm which uses code vectors to represent corner geometry information for searching the similar images from a database. The corners have been extracted by finding the intersections of the detected lines found using Hough transform. Taking the corner as the center coordinate, the angles of the intersecting lines are determined and are represented using code vectors. A code book has been used to code each corner geometry information and indexes to the code book are generated. For similarity measurement, the histogram of the code book indexes is used. This result in a significant small size feature matrix compared to the algorithms using color features. Experimental results show that use of code vectors is computationally efficient in similarity measurement and the corners being noise invariant produce good results in noisy environments.

Key words: CBIR, content based image retrieval, code vector, image features, image retrieval

Ⅰ. Introduction

Recent years have seen a rapid increase in the size of digital image collections. Everyday, both military and civilian equipment generates giga-bytes of images. A huge amount of information is out there. However, we cannot access or make use of the information unless it is organized so as to allow efficient browsing, searching, and retrieval. Content-based image retrieval (CBIR), a technique which uses visual contents to search images from large scale image databases according to users' interests, has been an active and fast advancing research area since the 1990s. During the past decade, remarkable progress has been made in both theoretical research and system development. However, there remain many challenging research problems that continue to attract researchers from multiple disciplines.

Feature (content) extraction is the basis of content-based image retrieval system. Image feature selection is critical because it largely affects the remaining aspects of the system design, and greatly

† 이 논문은 2008년도 조선대학교 학술연구비의 지원을 받아 연구되었습니다.
* 주저자 : 조선대학교 정보통신학과 박사수료
** 공저자 : 조선대학교 정보통신학과 석사과정
*** 공저자 : 조선대학교 정보통신학과 교수
**** 공저자 : 전남대학교 의공학과 교수(교신저자)
***** 공저자 : 조선대학교 전자공학과 교수(교신저자)
† 논문접수일 : 2008년 10월 21일
† 논문심사일 : 2008년 12월 8일(1차), 2009년 6월 22일(2차)
† 게재확정일 : 2009년 6월 25일

determines image retrieval capabilities of the eventual system. In a broad sense, features may include both text-based features (key words, annotations) and visual features (color, texture, shape, faces). However, since there already exists rich literature on text-based feature extraction in the DBMS and information retrieval research communities, we will confine ourselves to the techniques of visual feature extraction. Within the visual feature scope, the features can be further classified as general features and domain-specific features. The former include color, texture and shape features while the latter is application dependent and may include, for example, human faces and finger prints. The domain-specific features are better covered in pattern recognition literature and may involve much domain knowledge. After matching, images are ordered with respect to the query image according to their similarity measure and displayed for viewing [1].

Geometric shapes and corners form a major paradigm in the evaluations and identification of graphical information by brain (human perception) [2]. 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 code vectors for representing image features i.e., the corner geometry for image indexing and 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

Recently, many researchers have shown great interests in image retrieval based on compressed-domains such as vector quantization (VQ) [3-5], DCT [6] and DWT [7]. Reference [3] extracts features directly from the codeword indices of each spatial-domain VQ compressed image. Reference [4] extracts features from the individual codebook generated from each image. Reference [5] uses the index histogram of VQ-compressed index sequence to describe the features. Reference [6] obtains the texture features directly from the middle and low-frequency DCT coefficients. Reference [7] retrieves the images in 3 progressive steps using four lowest resolution subbands based on DWT. In [8] Rickman and Stonham proposed a color tuple histogram approach. They first constructed a code book which described every possible combination of coarsely quantized color hues that might be encountered within local regions in an image. Then a histogram based on quantized hues was constructed as the local color feature.

There is an abundance of literature on corner detection. Many researchers have adopted various approaches in finding the corners in images. Here research carried out on corner detection has been discussed briefly.

The Hough transform [9] later introduced in generalized form for lines and curve detection [10] has been focus of research interest after it was popularized by the journal article by D.H. Ballard [11]. Davies [12] applied the generalized Hough transform to corner detection. Diou, A. et al.[13] proposed an analytical approach for the calculation of the theoretical Hough transform on standard images for research on straight lines. Anastasios & Nikos [14] proposed the Inverse Hough Transform. Fei Shen & Han Wang [15] used modified Hough transform for corner detection. Yu-Hua Gu [16] 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 algorithm

In the proposed algorithm, we use the corner geometry information i.e., corner and intersecting line angles, for image retrieval. Corners at abstract level are the intersection point of curves or straight lines and have been defined in many ways in the literature. For the purpose of our algorithm and as a methodology for obtaining corner points in an image 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.

Hough transform has been used to search the straight lines in the images [10] using the parameterized line equation (1).

$$\rho = x\cos\theta + y\sin\theta \tag{1}$$

Each line in the image can be associated with a couple (ρ, θ) which is unique if $\theta \in [0, \pi]$ and $\rho \in R$, or if $\theta \in [0.\ 2\pi]$ and $\rho \geq 0$. 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 [14].

Ideally, a corner is an intersection of two straight lines. However, 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. 1). A corner can be characterized as one of the following four types:

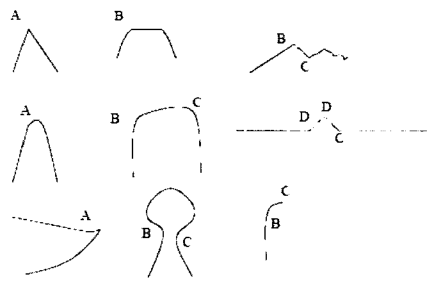
- Type A: A perfect corner as modeled in [17], i.e., a sharp turn of curve with smooth parts on both sides.
- Type B: The first of two connected corners similar to the END or STAIR models in [17], i.e., a mark of change from a smooth part to a curved part.
- Type C: The second of two connected corners, 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 1 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 or more straight lines.

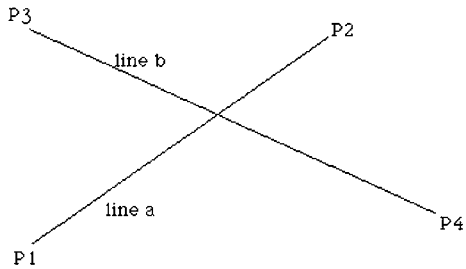
Figure 2 shows two intersecting line segments with given end point coordinates. The intersection point of the line segments can be computed as follows:

The equations of the lines are:

$$P_a = P1 + u_a(P2 - P1)$$
$$P_b = P3 + u_b(P4 - P3) \tag{2}$$



<Fig. 1> Four types of corners



<Fig. 2> Two intersecting line segments

Solving for the point where Pa=Pb gives the following two equations in two unknowns (ua and ub).

$$\begin{aligned} x1 + ua(x2 - x1) &= x3 + ub(x4 - x3) \\ y1 + ua(y2 - y1) &= y3 + ub(y4 - y3) \end{aligned} \quad (3)$$

Solving gives the following expressions for ua and ub.

$$\begin{aligned} u_a &= \frac{(x4-x3)(y1-y3)-(y4-y3)(x1-x3)}{(y4-y3)(x2-x1)-(x4-x3)(y2-y1)} \\ u_b &= \frac{(x2-x1)(y1-y3)-(y2-y1)(x1-x3)}{(y4-y3)(x2-x1)-(x4-x3)(y2-y1)} \end{aligned} \quad (4)$$

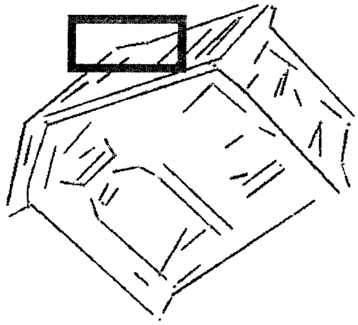
Substituting either of these into the corresponding equation for the line gives the intersection point. If the denominator for the equations for ua and ub is 0 then the two lines are parallel. If the denominator and numerator for the equations for ua and ub are 0 then the two lines are coincident. There are other cases also, such as if point of intersection lies on the projected 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.

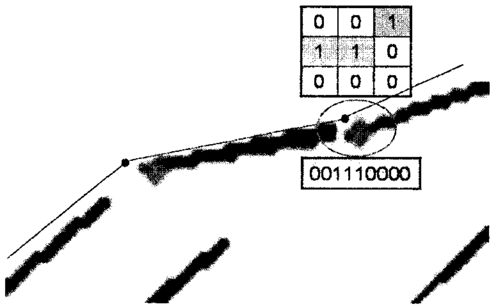
1. Feature Extraction and representation

Once the lines and corner information is obtained, lines intersection angle and the angles which the intersecting lines make with the x-axis are found. A 3x3 matrix has been used to represent the angles. The centre cell of the matrix being the corner and origin, each cell can represent intersecting lines within a range of 45 degrees. A '1' in the cell represents a line and '0' for the absence of a line. This information can be written using '9' binary digits (bits) corresponding to the 3x3 matrix.

Figure 3(a) below shows the detected lines in an image. The red rectangle shows the selected corners shown in the enlarged view in (b). The geometry of the



(a)



(b)

<Fig. 3> Corner Geometry

circled corner has been represented in the 3x3 matrix to show the angles of intersecting lines. The information can be written in binary form as '001110000'. Similarly corner geometry of each detected corner in an image can be represented using 9 binary digits.

Figure 4 shows few corner patterns and their binary representations.

1 0 0	0 1 0	1 0 1	0 0 0
0 1 0	1 1 0	0 1 0	0 1 1
1 0 0	0 0 0	0 0 0	0 1 0
100010100	010110000	101010000	000011010
0 1 0	0 1 0	0 1 0	0 0 0
0 1 0	0 1 0	0 1 1	1 1 0
1 0 0	0 0 1	0 0 0	0 0 1
010010100	010010001	010011000	000110001

<Fig. 4> Corner geometry representation

2. Code Book generation

A code book can easily be constructed offline, consisting of all the possible combinations of the intersecting lines, because of the binary representation of the corner geometry. Being a 9 digit binary number, all the possible combinations are from 000000000 ~ 111111111. By eliminating all the undesired entries, the only remaining elements in code book are 255. So the code book represents 255 corner signatures for image coding.

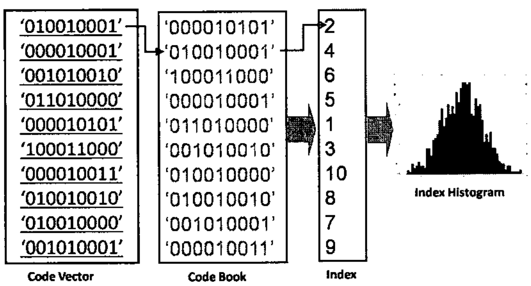
3. Feature Matrix generation

Code Book is searched for each code vector to find the index of the code book entry corresponding to the code vector. This gives the corner signature information of the image which is used to generate a histogram. The histogram of each image is stored in the feature database as feature matrix and used for comparing the similarity between the query image and database images.

Figure 5 shows the steps involved in generating a feature matrix as described above.

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



<Fig. 5> Generation of code book index

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.

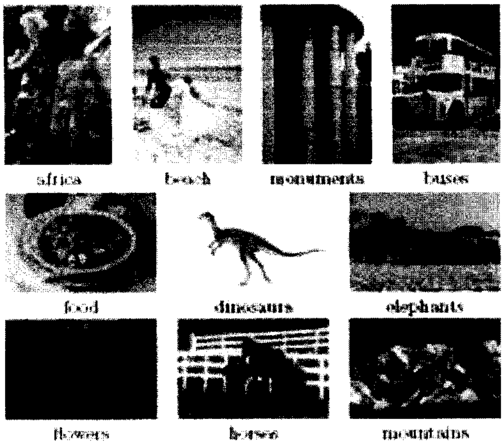
Let D be the image database and Q be the query image. The distance measure between D and Q, has been taken from [18]. We obtain a permutation of the images in D based on Q i.e., assign rank (I) $rank(I) \in |D|$ for each $I \in D$, using some notion of similarity to Q. This problem is usually solved 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:

$$|I - I'| = \frac{\sum |A_i - B_i|}{\sum |1 + A_i + B_i|} \qquad (5)$$

1. Data set

For testing the proposed idea, we used the Wang dataset [19]. The database is a subset of 1,000 images of the Corel stock photo database which have been manually selected and which forms 10 classes of 100 images each. Figure 6 shows different categories of the wang data set.

Figure 7 show a query image and the results obtained using the similarity measure discussed above. 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

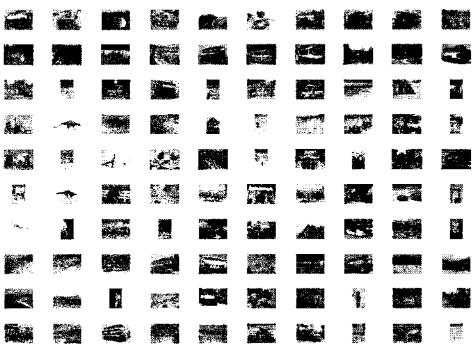


<Fig. 6> Categories of wang dataset

assumption that similar images will have similar corner signatures.

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.

(a) Query image



(b) Query results

<Fig. 7> Image retrieval results

V. Performance Evaluation

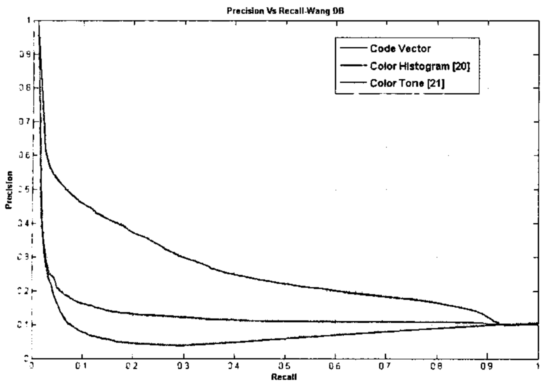
Two traditional measures for retrieval performance in the information retrieval literature are precision and recall. Precision is defined as the percentage of retrieved images that are actually relevant

$$Precision = \frac{Retrieved\ \&\ Relevent}{Total\ \#\ of\ retrieved} \quad (6)$$

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

$$Recall = \frac{Retrieved\ \&\ Relevent}{Total\ \#\ of\ relevant} \quad (7)$$

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 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. Figure 8 shows the averaged precision and recall for the entire database compared with two other color based algorithms [20, 21].



<Fig. 8> Performance evaluation

Experiments show that feature code vector has a better precision in retrieval.

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.

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. 2000.

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

[3] S. Panchanathan and C. Huang, "Indexing and retrieval of color images using vector quantization," *SPIE Proc. Applications of Digital Image Processing XXII*, 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, Jun. 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. 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, February 1996.

[9] P. V. C. Hough, "Method and means for recognising complex patterns," U. S. Patent no. 3069654, Dec. 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. 11-15, Jan. 1972.

[11] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111-122, Apr. 1981.

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

[13] A. Diou, Y. Voisin, and C. Santo, "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," Pattern Recognition Letters vol. 23, no. 8, pp. 1039-1049, June 2002.

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

[17] 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.

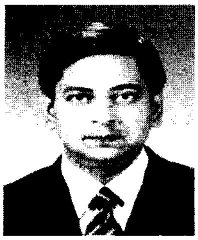
[18] J. Huang, S. R. Kumar, M. Mitra, W. J. Zhu, and R. Zabi, "Image indexing using color correlograms," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 762-768, June 1997.

[19] J. Li and J. Z. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1075-1088, Sept. 2003.
http://wang.ist.psu.edu/docs/related.shtml

[20] J. A. Park, S. J. Han, and Y. E. An, "Heuristic features for color correlogram for image retrieval," *Proc. Int. Conf. Computational Sciences and Its Applications*, pp. 9-13, July 2008.

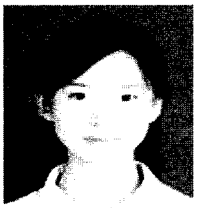
[21] Y. E. An, S. B. Pan, and J. A. Park "Image retrieval based on color tone variance difference feature," *Proc. Int. Conf. Machine Learning and Cybernetics*, vol. 7, pp. 3777-3780, July 2008.

저자소개



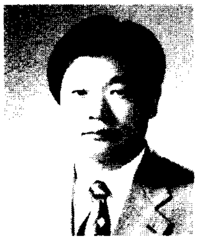
Nishat, Ahmad

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



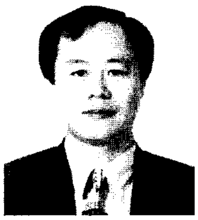
조 혜 (Zhao Hui)

2007년 : Nanjing University of Information Science and Technology 공학사
2008년 ~ 현재 : 조선대학교 정보통신공학과 석사과정



박 종 안 (Park, Jong-An)

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



박 승 진 (Park, Seung-Jin)

1997년 ~ 현재 : 전남대학교 병원 의공학과 교수
1991년 ~ 1997년 : 전남대학교 경상대학교 보건사무관
1991년 : 조선대학교 전기공학과(전자전공) 공학박사
1986년 : 조선대학교 전자공학과 공학석사
1984년 : 조선대학교 전자공학과 공학사

양 원 일 (Yang, Won-Il)

1983년 ~ 현재 : 조선대학교 전자공학과 교수
2001년 : 동신대학교 전기전자공학과 공학박사
1982년 : 조선대학교 전기공학과 공학석사
1972년 : 조선대학교 전기공학과 공학사