Building Plan Reconstruction based on codebook and keyblock framework

I Gusti Agung Gede Arya Kadyanan[1], Nyoman Gunantara[2], Ida Bagus Gede Manuaba[3], and Komang Oka Saputra[4]

[1,2,3,4] Udayana University, Jimbaran Badung 80361, Indonesia
gungde@unud.ac.id

Abstract. These days, one of the biggest challenges facing pattern recognition research is building plan recognition. The findings of an exploratory investigation on *keyblock* framework-based building plan recognition are presented in this paper. This study also aims to demonstrate the universal applicability of the *keyblock* framework for CBIRS. Using the *keyblock* features for encoding and decoding building plan images, a *codebook* is constructed in this research. This study uses the Generalized Lloyd Algorithm (GLA) to extract pertinent data and minimize block during *codebook* generation. A one-dimensional matrix representation of the encoding image is used, which is comparable to a set of keywords in a text retrieval system. To reconstruct and query the image using an example, we performed image decoding. The effectiveness of the *keyblock* framework on building plan reconstruction was assessed using the root-mean-square error of the rebuilt image as the performance metric.

Keywords: *Keyblock*, Vector Quantization, *Codebook*, Generalized Lloyd Algorithm, Building Plan Reconstruction

1 Introduction

For the past few years, the recognition of building plans has remained an unresolved issue. These days, digital photographs are quite significant. The growing application of building plan recognition across a range of domains, including site mapping and location determination from satellite data, reflects this. Images hosted on the service have been accessed and altered by users with a variety of backgrounds [1]. Because of the complexity of picture understanding and the wide range of applications, system queries and content-based image database browsing may not always yield satisfactory results. Additionally, the system must be able to meet the requirements for accuracy and efficiency. These days, image retrieval systems based on content are becoming a major topic of study.

By comparing input photographs with building plans kept in a database, building plan recognition is used to identify one or more people from a collection of images or a video image sequence of a scenario. One advantage of building plan recognition is

K. Dharmawan and N. A. Sanjaya ER (eds.), *Proceedings of the First International Conference on Applied Mathematics, Statistics, and Computing (ICAMSAC 2023)*, Advances in Computer Science Research 110,
https://doi.org/10.2991/978-94-6463-413-6_5

that it is a passive, non-intrusive method of naturally and amiably verifying an individual's identification [9].

A common query technique is "query by example," in which the user submits an example image of a human building plan to the system, and it returns a list of related human building plan photos from its database. Features related to color, texture, and shape are used to define the similarity measurements [2]. The efficacy of obtaining pertinent retrieved images of human construction plans is the real subject of this research on the content-based image retrieval system.

This study focused on human building plans. characteristic of texture. Since texture is one of the primary elements of human visual perception, texture features play a significant role in retrieval systems. Even though it is more difficult to characterize texture features than other features, humans can nevertheless recognize textures with ease. Texture, as opposed to color feature, specifies a region characteristic as opposed to merely a pixel characteristic. Among other things, directionality, roughness, regularity, and contrast can be described by texture features. Texture features are often used in the feature extraction and selection process.

A preliminary analysis of building plan texture features based on the keyblock framework is presented in this research. Images are encoded and decoded using a *codebook*. A one-dimensional matrix representation of the encoding image is used, which is comparable to a set of keywords in a text retrieval system. We performed picture decoding to reconstruct the encoded image, and we computed the resultant image's root-mean-square error to assess the effectiveness of the keyblock architecture in reconstructing the architectural design. A total of 121 human building plan photos, 73 images for query testing, and 48 images for training data were employed in this MATLAB software-based experiment.

2 Texture Features

Texture is a crucial component of natural objects, including textiles, plants, and architectural blueprints. Texture provides details about a building plan's structural layout and how it interacts with the surroundings. A human can identify texture with ease, while a digital computer finds it challenging to define.

A digital image can be represented as a two-dimensional array. Let $Lx = \{1, 2, ..., Nx\}$ and $Ly = \{1, 2, ..., Ny\}$ be spatial data, hence the digital image I has a spatial resolution of Lx x Ly or has Lx x Ly resolution cells (pixels). An image I is a function that maps gray levels $G \in \{1, 2, ..., Ng\}$ to each resolution cell, $I: Lx \times Ly \rightarrow G$. Gray levels show the variations of darkness and lightness of an image cell, while texture values represent the statistics or the spatial distribution of gray levels.

Although they can both be dominant, texture values and gray level spatial distribution have a strong relationship. The main property in an image is its gray level, even though a short part of the image has relatively little fluctuation in its gray level values. Texture takes center stage when there is a greater variance in the gray level values [3]. The spatial pattern of a gray-level-pixel area is a crucial aspect of a texture characteristic. The texture is smooth if the gray-level range is broad and there is no spatial pattern. A texture is considered rough if it has a distinct spatial pattern and a higher number of gray levels. You may get many kinds of textures in [4].

3 *Keyblock* Framework

The use of keywords and a dictionary has proven successful in text-based retrieval system applications on Yahoo, Lycos, and Google. L. Zhu et al. [5] first presented an analogous method based on a *codebook* and *keyblock* for image-based retrieval systems in 2002. A picture is separated into image segments (blocks) of the same size. In order to identify *keyblocks* and create a *codebook* that represents each block of every image in an image database, vector quantization clustering (VQ) techniques such as Generalized Lloyd Algorithm (GLA) and Pair Wise Nearest Neighbor Algorithm (PNNA) are typically utilized. A block diagram of the feature extraction procedure and picture retrieval based on keyblock is displayed in **Fig. 1** [5].

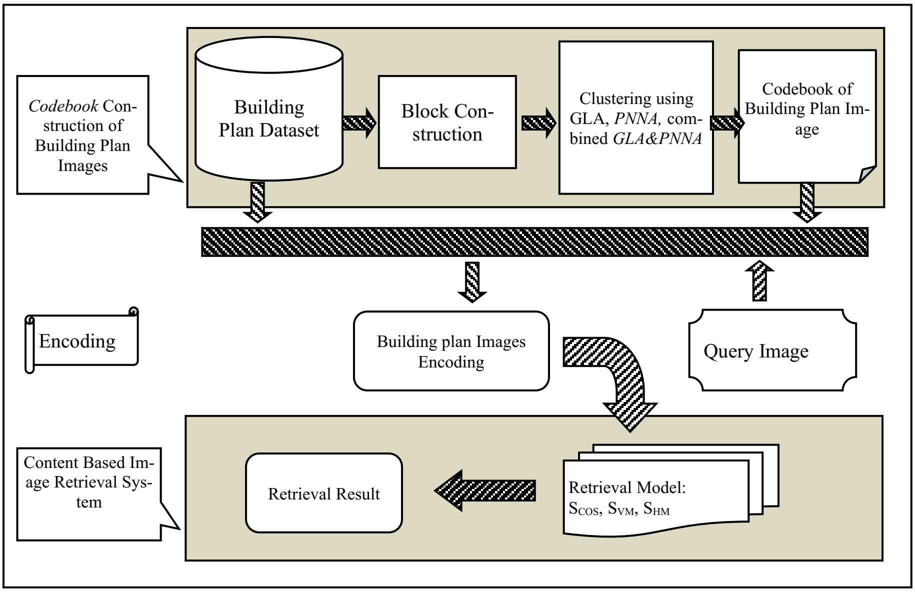

Fig. 1. A block diagram of images extraction and retrieval process based on *keyblock* [5].

A sample set of training images has been chosen from the database. Block sizes for images are selected. *Keyblocks* are identified using the GLA to create a *codebook* based on the blocks obtained from the training image sample sets. Furthermore, using the generated *codebook*, every image in the database may be recreated.
An image that needs to be rebuilt is broken down into individual image blocks. Based on the entry index of the most similar keyblock in the *codebook*, an index is assigned

to each block in the image. This allows a feature vector to be used to represent an image, with each feature value being associated with the most similar block in the *codebook*. Ultimately, a low dimensional feature vector that refers to the keyblock index in the *codebook* can replace every image in the database. An image's feature vector can be used to approximate a reconstruction of the image (see Fig. 2). It is now possible to extend the idea of a *codebook* and *keyblocks* framework in the image-based retrieval system to a dictionary and keywords in the text-based retrieval system [6].

4 Generalized Lloyd Algorithm (GLA)

GLA is a clustering method that is used iteratively to create the best vector quantifier design. The following procedures are followed in every phase of the iteration. The closest entry in the *codebook*, which was acquired during the preceding iteration phase, is used to cluster the training vector sample set. Based on the clustering outcome, all centroids' values are recalculated. Each centroid that is obtained during the iteration stage will be included in the *codebook*. Every distortion has been calculated and assessed. An additional iteration is necessary if the distortions continue to be greater than a predetermined threshold; otherwise, the clustering process is stopped.

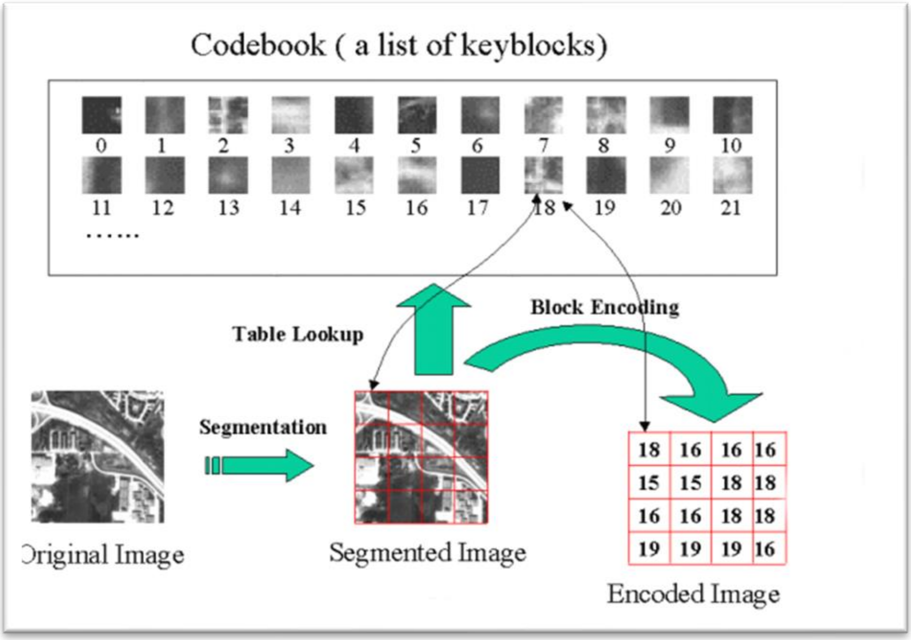

Fig. 2. Procedure of image encoding and decoding

The followings are the GLA steps as explained in [8]:

- Step 1 (initialization)
  – Choose training set $T = \{t_1, ..., t_l\}$ randomly;

- Set a threshold value δ;
- Initial *codebook*, $C_l = \{c_1, ..., c_i, ..., c_N\}$;
- Initial average distortion, $D_0 = \infty$ (a large value); and
- Set the iteration count, $m = 1$.

- Step 2 (clustering iteration)
  Use C_m to find the minimum distortion partition $P = \{P_1, ..., P_i, ..., P_N\}$ of T based on the nearest neighbor condition, and compute the average distortion D_m

$$D_m = \frac{1}{l} \sum_{i=1}^{N} \sum_{t \epsilon Pi} d(t, cj). \tag{1}$$

- Step 3 (distortion calculation)
  If $\frac{D_{m-1} \quad - D_m}{Dm} \leq \delta$, then terminate the iteration with the C_m as the final *codebook*, otherwise continue.
- Step 4 (update *codebook*)
  Generate new *codebook Cm+1* by computing the centroid of each partition P;
  $c_i = \frac{1}{|P_i|} \sum_{t \epsilon P_i} \mathrm{t}$
  and replace the old *codebook Cm* with new *codebook Cm+1*, set $m = m+1$ and go to step 2.

5 Root Mean Square error (RMS_error)

We can reconstruct the entire image of the training results of the encoding process to demonstrate the correctness of the resultant *codebook*. We then use the method to compute the intensity of gray RMS error value that exists between the reconstructed image and the original image.:

$$RMS_{error} = \sqrt{\frac{1}{N_1 N_2} \sum_{N_1=0}^{N_1-1} \sum_{n_2}^{N_2-1} (x[n_1, n_2] - \hat{x}[n_1, n_2])^2} \tag{2}$$

Where;

N_1 : Row Size
N_2 : Column Size
n_1 : Rows Coordinate
n_2 : Column Coordinate

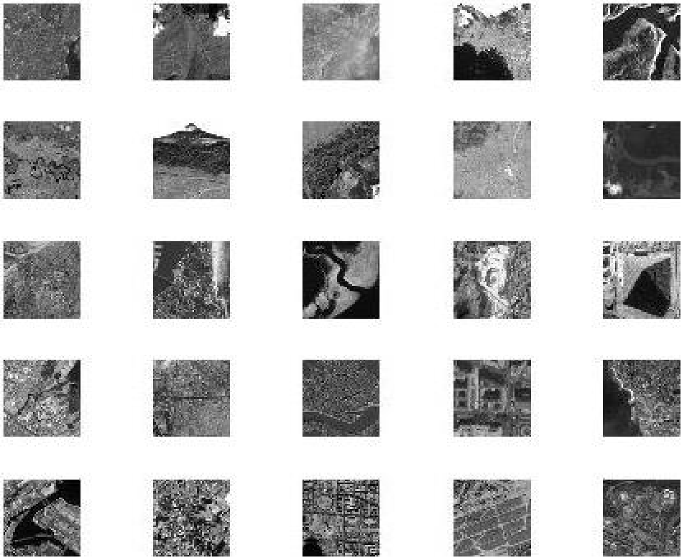

Fig. 3. Typical 25 of 121 original building plan images from satellite imagery

6 Experimental Results and Analysis

A total of 121 building plan photos were used in this initial investigation. The building plan pictures have a resolution of 128 by 192. The greatest performance block size for recognizing building plans is determined by a number of block sizes, including 2 by 2, 4 by 4, and 8 by 8. There are 100, 200, 300, 400, and 500 entries in the *codebook*. All photos stored in the database undergo image encoding and decoding. 25 of 121 original building plan images are displayed in Fig. 3. The reconstructed images based on the matching *codebook* are displayed in Figures 4 through 6. The outcome of the experiment demonstrates that the vector quantifier for image encoding and decoding can be the *codebook*. The root-mean-square error (RMS) value at the average gray level is used to gauge the quality of the reconstructed images. For the reconstructed images displayed in **Table 1**, the ideal gray level RMS errors value is 5.93.

Table 1. *RMS_error* average value

Block Size	***Codebook Size***				
	100	200	300	400	500
2 x 2	6,85	6,71	6,29	6,04	5,93
4 x 4	8,12	8,08	7,92	7,78	7,75
8 x 8	8,45	8,60	8,43	8,47	8,50

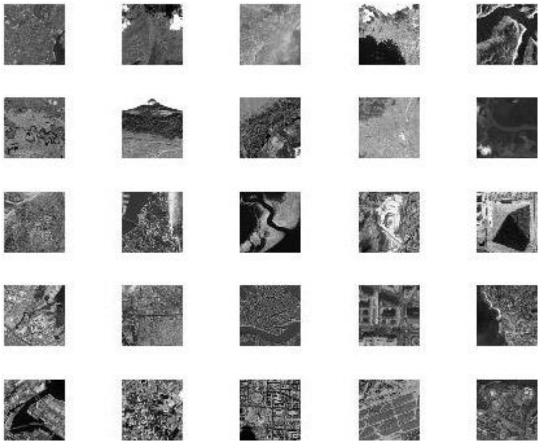

Fig. 4. Image reconstruction using block size of 2 by 2 and *codebook* size of 100.

Fig. 5. Image reconstruction using block size of 4 by 4 and *codebook* size of 100.

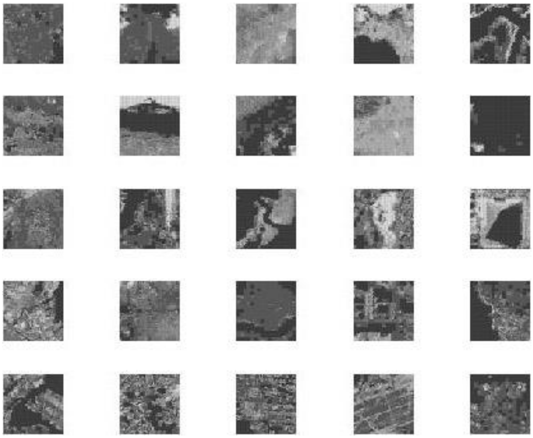

Fig. 6. Image reconstruction using block size of 8 by 8 and *codebook* size of 100.

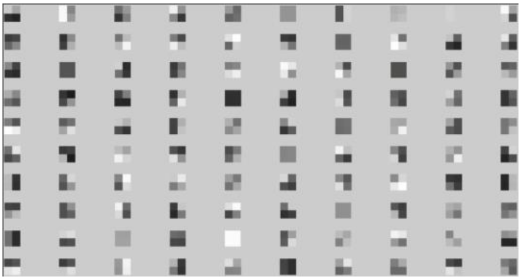

Fig. 7. Codebook using block size of 2 by 2 and codebook size of 100

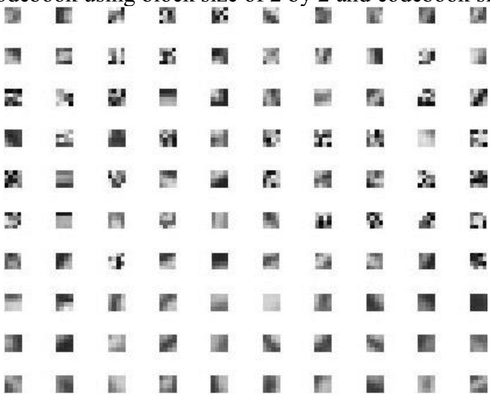

Fig. 8. Codebook using block size of 4 by 4 and codebook size of 100

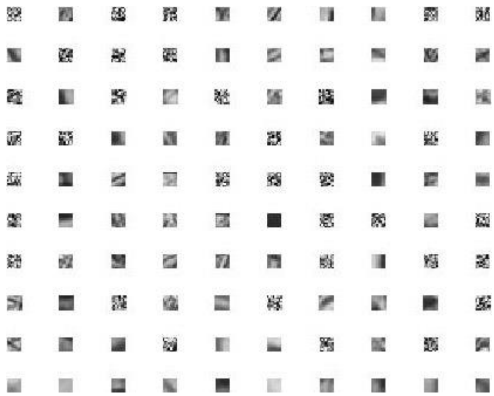

Fig. 9. Codebook using block size of 8 by 8 and codebook size of 100

The acquisition performance of a *codebook* tends to rise as its size increases, inversely proportionate to the larger *codebooks* decreasing average RMS_error. From the two aforementioned points, it can be observed that the average RMS_error value can be used to determine the *codebook* and block sizes for the image acquisition system's effectiveness, with smaller RMS_error values indicating higher performance.

7 Concluding Remarks

An initial investigation has been conducted into the development of an effective database for building plan images, utilizing a *codebook* and *keyblocks* framework for image encoding and decoding. The best experimental findings demonstrate that, for block sizes of 2 by 2 and *codebooks* of 500, it is possible to reconstruct the approximated images from the compressed image data using the framework with gray level RMS error values between 5.93 and 6.85. Additionally, the experimental results demonstrate that when blocksize increases, the average gray level RMS error tends to grow dramatically.

8 Acknowledgment

The development of content-based image retrieval system for building plan recognition was supported by Image Processing and Pattern Recognition laboratory in faculty of mathematic and natural science Udayana University.

References

1. J. Eakins and M. Graham, “Content-based Image Retrieval,” *Technology Applications Programme Report 39*, University of Northumbria at Newcastle, October 1999.
2. http://en.wikipedia.org/wiki/CBIR, “Content-based image retrieval” [accessed in October 1, 2009].

3. R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Feature for Image Classification," *IEEE Transactions on System, Man and Cybernetics*, Vol SMC-3 No. 6, November 1973.
4. P. Brodatz, "Textures", New York, Dover, 1966.
5. L. Zhu and A. Zhang, "Theory of *Keyblock*-based Image Retrieval," *ACM Journal*, pp. 1-32, March 2002.
6. L. Zhu, A. Rao, and A. Zhang, "Advanced Feature Extraction for *Keyblock*-based Image Retrieval", *Proceeding of International Workshop on Multimedia Information Retrieval (MIR 2000)*, Los Angeles, California, USA, November 4, 2000.
7. L. Zhu, A. Rao, and A. Zhang, 2000, "*Keyblock*: An approach for content-based geographic image retrieval," *Proceedings of First International Conference on Geographic Information Science (GIScience2000)*, Savannah, Georgia, USA, 286–287, 2000.
8. L. Zhu, C. Tang, and A. Zhang, "Using Key block Statistics to Model Image Retrieval," Advances in Multimedia Information Processing – PCM 2001, Second IEEE Pacific Rim Conference on Multimedia, Beijing, China, October 24-26, 2001, Proceedings 2001.
9. Gao, Y., K.H. Leung M. "Building plan Recognition Using Line Edge Map" IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.24, NO.6, June 2002.